Counteracting Phishing through HCI: Detecting Attacks and Warning Users

Dissertation

an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt von
Diplom-Medieninformatiker

Max-Emanuel Maurer

München, den 15. Dezember 2013

Counteracting Phishing through HCI: Detecting Attacks and Warning Users

Dissertation

an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt von
Diplom-Medieninformatiker
Max-Emanuel Maurer

München, den 15. Dezember 2013

Erstgutachter:	Prof. Dr. Heinrich Hußmann
Zweitgutachter:	Prof. Dr. Marc Langheinrich

Tag der mündlichen Prüfung: 11. April 2014

ABSTRACT

Computer security is a very technical topic that is in many cases hard to grasp for the average user. Especially when using the Internet, the biggest network connecting computers globally together, security and safety are important. In many cases they can be achieved without the user's active participation: securely storing user and customer data on Internet servers is the task of the respective company or service provider, but there are also a lot of cases where the user is involved in the security process, especially when he or she is intentionally attacked. Socially engineered phishing attacks are such a security issue were users are directly attacked to reveal private data and credentials to an unauthorized attacker. These types of attacks are the main focus of the research presented within my thesis.

I have a look at how these attacks can be counteracted by detecting them in the first place but also by mediating these detection results to the user. In prior research and development these two areas have most often been regarded separately, and new security measures were developed without taking the final step of interacting with the user into account. This interaction mainly means presenting the detection results and receiving final decisions from the user. As an overarching goal within this thesis I look at these two aspects united, stating the overall protection as the sum of detection and "user intervention".

Within nine different research projects about phishing protection this thesis gives answers to ten different research questions in the areas of creating new phishing detectors (phishing detection) and providing usable user feedback for such systems (user intervention): The ten research questions cover five different topics in both areas from the definition of the respective topic over ways how to measure and enhance the areas to finally reasoning about what is making sense. The research questions have been chosen to cover the range of both areas and the interplay between them. They are mostly answered by developing and evaluating different prototypes built within the projects that cover a range of human-centered detection properties and evaluate how well these are suited for phishing detection. I also take a look at different possibilities for user intervention (e.g. how should a warning look like? should it be blocking or non-blocking or perhaps even something else?). As a major contribution I finally present a model that combines phishing detection and user intervention and propose development and evaluation recommendations for similar systems. The research results show that when developing security detectors that yield results being relevant for end users such a detector can only be successful in case the final user feedback already has been taken into account during the development process.

ZUSAMMENFASSUNG

Sicherheit rund um den Computer ist ein, für den durchschnittlichen Benutzer schwer zu verstehendes Thema. Besonders, wenn sich die Benutzer im Internet – dem größten Netzwerk unserer Zeit – bewegen, ist die technische und persönliche Sicherheit der Benutzer extrem wichtig. In vielen Fällen kann diese ohne das Zutun des Benutzers erreicht werden. Datensicherheit auf Servern zu garantieren obliegt den Dienstanbietern, ohne dass eine aktive Mithilfe des Benutzers notwendig ist. Es gibt allerdings auch viele Fälle, bei denen der Benutzer Teil des Sicherheitsprozesses ist, besonders dann, wenn er selbst ein Opfer von Attacken wird. Phishing Attacken sind dabei ein besonders wichtiges Beispiel, bei dem Angreifer versuchen durch soziale Manipulation an private Daten des Nutzers zu gelangen. Diese Art der Angriffe stehen im Fokus meiner vorliegenden Arbeit.

Dabei werfe ich einen Blick darauf, wie solchen Attacken entgegen gewirkt werden kann, indem man sie nicht nur aufspürt, sondern auch das Ergebnis des Erkennungsprozesses dem Benutzer vermittelt. Die bisherige Forschung und Entwicklung betrachtete diese beiden Bereiche meistens getrennt. Dabei wurden Sicherheitsmechanismen entwickelt, ohne den finalen Schritt der Präsentation zum Benutzer hin einzubeziehen. Dies bezieht sich hauptsächlich auf die Präsentation der Ergebnisse um dann den Benutzer eine ordnungsgemäße Entscheidung treffen zu lassen. Als übergreifendes Ziel dieser Arbeit betrachte ich diese beiden Aspekte zusammen und postuliere, dass Benutzerschutz die Summe aus Problemdetektion und Benutzerintervention' („user intervention“) ist.

Mit Hilfe von neun verschiedenen Forschungsprojekten über Phishingschutz beantworte ich in dieser Arbeit zehn Forschungsfragen über die Erstellung von Detektoren („phishing detection“) und das Bereitstellen benutzbaren Feedbacks für solche Systeme („user intervention“). Die zehn verschiedenen Forschungsfragen decken dabei jeweils fünf verschiedene Bereiche ab. Diese Bereiche erstrecken sich von der Definition des entsprechenden Themas über Messmethoden und Verbesserungsmöglichkeiten bis hin zu Überlegungen über das Kosten-Nutzen-Verhältnis. Dabei wurden die Forschungsfragen so gewählt, dass sie die beiden Bereiche breit abdecken und auf die Abhängigkeiten zwischen beiden Bereichen eingegangen werden kann. Die Forschungsfragen werden hauptsächlich durch das Schaffen verschiedener Prototypen innerhalb der verschiedenen Projekte beantwortet um so einen großen Bereich benutzerzentrierter Erkennungsparameter abzudecken und auszuwerten wie gut diese für die Phishingerkennung geeignet sind. Außerdem habe ich mich mit den verschiedenen Möglichkeiten der Benutzerintervention befasst (z.B. Wie sollte eine Warnung aussehen? Sollte sie Benutzerinteraktion blockieren oder nicht?). Ein weiterer Hauptbeitrag ist schlussendlich die Präsentation eines Modells, dass die Entwicklung von Phishingerkennung und Benutzerinteraktionsmaßnahmen zusammenführt und anhand dessen dann Entwicklungs- und Analyseempfehlungen für ähnliche Systeme gegeben werden. Die Forschungsergebnisse zeigen, dass Detektoren im Rahmen von Computersicherheitsproblemen die eine Rolle für den Endnutzer spielen nur dann erfolgreich entwickelt werden können, wenn das endgültige Benutzerfeedback bereits in den Entwicklungsprozesses des Detektors einfließt.

ACKNOWLEDGMENTS

The four years of doing a PhD are a long period of time. A lot of different research is conducted in that period, one learns a lot of things even besides the own thesis topic. Perhaps the most important of those side issues is the fact that I met a lot of bright people during these four years that inspired, influenced and helped me in the context of this thesis and that will influence my life forever. Listing them all would most probably fill the whole of this thesis with acknowledgments leaving no room for the research contents but at least a few have to be named in the following sentences.

The two most important people for this thesis naturally are my two supervisors Professor Dr. **Heinrich Hußmann** and Professor Dr. **Marc Langheinrich**. Prof. Hußmann inspired me from the moment I started at the university first as Professor, later as a boss and colleague and finally also being my supervisor. His abilities to immerse himself in other people's problems within minutes and give extraordinary and well understandable feedback fascinated me in many of our discussions about this thesis. Despite the many different research paths his PhD students take he has the right insights, inputs and sources for all of them. The level of detail of feedback he provided for this thesis was exceptional and I want to thank him especially for these vast amounts of time that he spent helping me. A last important thing to thank him for is the term "user intervention" that he coined in the way that it will be used within this thesis. Besides the great support I received by my internal supervisor, I also met Prof. Langheinrich already early in my PhD career and was always fascinated of his friendly and open character that led to many interesting discussions. Organizing several workshops together with him, Rene Mayrhofer and Alexander De Luca was always and immense pleasure. After becoming my second external supervisor I was sadly not able to visit him at his lab in Lugano, yet. I am really looking forward to more collaboration in the future.

I will always remember the unbelievably kind colleagues I had the pleasure to work with and hope that we will meet more often in the near future. Having so many different teams and people in our two research groups – the second one being led by Professor Dr. **Andreas Butz** who also helped me a lot during my university times (regards to his family) – it seems extremely hard to structure those relationships: so let's try this by offices. From my office colleagues **Alexander De Luca** was the one who initially set me up with usable security research. In a way he is the father of my thesis direction and also the one who introduced me to Thai Karaoke. **Sara Streng** who is now a former colleague for a long time was the female part and managed to keep our lively office calm until one day she sadly left – making room for me at the window side – and **Simon Stusak** joined our office to replace me little by little as a student counselor. Thanks for taking care of most of my former duties. In the office next door **Alexander Wiethoff** showed me how neat research documentation can look like sharing the room with **Hendrik Richter** who managed to be a couple of months faster with his thesis although he did so many "haptic" projects: incredible. As in most offices the most underestimated role is the one of the office secretary and especially for a student counselor this person is even more important. Over the years we had different great people for this role but I want to give my special thanks to **Franziska Schwamb** who was always so caring and attentive and was thinking of everything that I missed. Continuing with the great colleagues from the second floor I have to mention **Doris Hausen** first. From the first semester onwards our paths at the university career were running parallel. Studying together for the exams she now also submitted her thesis a couple of weeks before mine. A long lasting friendship that will hopefully continue onwards long after our PhD period. She also was an important part of our UniWorX development team at the university. Besides the two of us it consisted of **Emanuel von Zezschwitz** – another one following the path of usable security and one of the most honest and nicest characters I met during my university time – and **Henri Palleis** – who managed to change the flash course I was teaching for several years making a worthy HTML5 successor. **Raphael Wimmer** who is a former member of the group with his unbroken interest to everything showed me many possible paths at the university after my start. **Alina Hang** – the always positive and smiling colleague, despite tough project times she had, is such a great character who hopefully will also find the final bit of her PhD quickly. Sharing the office with **Sebastian Löhmann** who inspired me with his great talent of planning and keeping a balanced schedule although the big project he took care of. Hopefully there will be the day when I find the time to follow all his running advice. The office of Alina and Sebastian was formerly owned by **Sebastian Boring** and **Dominikus Baur** who were the senior PhDs I could look up to and learn a lot from when I started at the university. Furthermore I want to thank **Fabian Hennecke** for the many fun and jokes but also the serious discussion partner that he could be and all the many excellent web links and articles that he found that were all relevant in my area. Last but not least our first floor hero who has been the best system administrator one could ever imagine **Rainer Fink** has to be named who knows where all precious research data would have gone if he would not have been there.

Another important group of people to mention are the many talented students that I supervised during my years at the university. Many of them wrote theses that were closely related to my PhD topic and are mentioned within the respective chapters of this thesis. They are also the reason why large parts of this thesis are written in the scientific plural. Thank you all for sharing my research interests and helping me bringing usable security research forward.

But all the wonderful colleagues and students are only one side of the coin of doing a PhD. The other side is the outstanding support I received by my family. My mother **Ingeborg** – who is sadly no longer among us – and my father **Maximilian** supported me so much through all the years as a student and as a PhD student and believed in my university career. Thank you both for this. Finally I want to thank my beautiful, loving and caring wife **Sonja** for being the best partner one could ever imagine and for giving birth to the newest and biggest part of my life so far my son **Max Leonard**.

TABLE OF CONTENTS

List of Figures

I

Introduction

Chapter 1

Introduction

Computer security is one of the oldest computer topics of all and with each new use for computers, new threats and security issues come up. When the Internet started to become popular and accessible for the masses at the beginning of the 1990s, security threats also started to evolve there (see section 2.4 for more details and references). One of the most popular of those threats forms one of the central aspects of this thesis: phishing. Phishing being a form of online identity theft using socially engineered attacks will be more closely explained and defined in section 2.1. Two aspects of the Internet form a fertile ground for this relatively new kind of attacks: more and more security related actions are carried out using it while the average technical knowledge of each user is getting less.

Phishing as a social engineering attack, in contrast to many other security threats, does not only make use of security flaws of software but uses a social component to trick the users into making a security mistake. Whilst phishing started off because some people wanted to get on the Internet for free and hence stole user account data to go online, phishing nowadays is much more focused, professional and mostly dedicated to monetary revenue, and it is still on the rise as just recently reported by Kaspersky Labs [146]. In a typical phishing attack a user receives an email claiming that something is wrong with one of her accounts and she needs to reenter the data. A website link is included in the email leading the user to a convincing looking website that is in fact a fake copy of the original website. In case the user enters her credentials on the website, the phishers get hold of them and can furthermore impersonate the victim and conduct transactions on her behalf.

This criminal act of stealing personal data is hard to impede especially by classic law enforcement. However, anti-virus companies, browser vendors and other Internet stakeholders tried to create software detectors that stop those attacks from happening. In case such an attack is found, the user usually gets some kind of notification about the threat and has to decide how to continue. These two main parts of detection and intervention are, what lies in the center of this thesis with a focus on usable security.

For some of the work within this thesis other researchers have been involved to a small extent. In many cases parts of the programming work or the execution of a user study have been conducted as part of a bachelor or masters thesis under my supervision. At the beginning of the respective project chapters all involved persons and resulting publications will be named. Except for literal quotations and referenced figures all content of this thesis has been created by myself.

1.1 Usable Security

Computer security measures in general are not ill-developed. Taking a 2048bit-SSL certificate[1] as an example, it would on average take 6.4 quadrillion years with a standard desktop computer to crack such a key [68] or in other words 6.4 quadrillion desktop computers would need one year. SSL certificates and encryption work also very well despite some minor attack possibilities [339]. Hence, the math for most computer security problems seems to be fine, but still a lot of security attacks towards computers, companies and users happen each day. In such cases security experts often like to blame the users as being "the weakest link in the chain" to shove away responsibility for a security issue: users disclose their passwords, fail to encrypt confidential information, switch virus checkers off and do so many more irrational things from a security perspective (see Sasse and Flechais [255] book chapter about usable security for more details).

But why does this happen? Asking this question already brings us right into the heart of usable security. Usable security is all about finding out why the user is the weakest link of the security chain and how this can be avoided.

Security and usability are two areas that contradict each other to some extent. Think of a door looked with 10 different locks. It will be safer than a door with only one lock but the persons that want to enter will have to do a lot more work. The challenge for usable security is to get security and usability into the right ratio. As Adams and Sasse already point out it is important not to blame the users if they try to circumvent too complex security measures. As a result of this, the design of security measures should be centered around them [3,331,338].

Social engineered phishing attacks on the Internet are one, if not the most prominent example for an area where usable security is extremely important. Bardzell et al. [21] elaborate on the human-centered design considerations for phishing in their book chapter which offers a very good starting point to familiarize oneself with this topic more closely. They explain some of the base assumptions that have to be made about users and online security and give some general recommendations for user-centered security development. To ease such

[1] A certificate in general binds a public key needed for encryption of a secure connection to other properties of the issuing company. Certificates can be signed by other parties to validate that a certificate is trusted by this party. Extended validation certificates contain more information about the issuer and have a more controlled validation process before they are signed by a certificate authority [39].

development other researchers have taken general usability criteria by Jakob Nielsen [216] and applied them to usable security [143].

1.1.1 Usable Warning Design

The design of computer warnings driven by usability can be seen as a smaller subfield of usable security in general that is of great importance for this thesis. Developing new warnings and evaluating their performance has been done with warnings in the physical world for years now (e.g. by Wogalter [316]).

Some of the findings of this research can be applied to computer warnings too, but computer warnings have special properties that set them apart from other warnings in the physical world. While the focus of classical warning research is more on properties like form, color, light, size, imagery or texture, these kinds of warnings are hard to personalize but easier to set apart from other information around them. On the other hand many properties of computer warnings are not changeable (e.g. light, texture), but the possibility of including personalized contents is one of their strengths. We will elaborate on this more closely in the related work section 3.3.

1.2 Problem Statement

Encrypting information up to a level that cannot ever be decrypted again using methods and technologies we know up to now, seems an easy task for security development but in many other cases developing security countermeasures for attacks seems to be much harder. Virus-checkers are supposed to find malicious software running on the computers of their victims but the virus checking software itself heavily relies on a given database of already known attacks that has to be updated over and over again. But what would happen if each virus appearing at each user would be a completely individual one? This would render current anti-virus software useless to large extents.

In the case of phishing, a similar counter-measure is used today. Malicious severs and URLs are put on a blacklist index and users being protected by this blacklist will not fall for a phishing site hosted under such a domain. While the approach still works somehow well for virus detection, phishers actually already have found ways to make each phishing attack unique for each user. This can render blacklists useless in the near future (a more detailed explanation of this will follow in section 3.2). In the end this means that the search for new detectors that are beyond standard blacklists is necessary as the loss of effect of current methods is imminent.

Research for such detectors is going on for several years now but as such detectors are usually unable to perfectly judge the tested websites the user is needed in a final stage to decide on the authenticity of the suspected website. With the user being the target of the attacks and the person who is in charge of finally judging attack attempts, the main goal of this thesis is to provide new insights on protection against phishing by putting the user and his behavior in the center of the considerations.

Besides the problem of the attacks themselves phishing is successful because of more general Internet properties. As a global medium it brings information and offers from all over the world right into the users' living room within milliseconds. On the other hand as fast as data is transmitted inbound as fast important security credentials are lost to a phisher outbound. These properties additionally go hand in hand with the problem that security is never the primary goal of a user [296] on the Internet. Users want to be secure without security being their major concern.

Looking at this problem space a lot of questions arise: how can attacks be detected? How many of them can be automatically found? Can warnings help to alert the user of this threat? How should those warnings look like? Can HCI and usable security help to maximize the power of such approaches?

This thesis and its projects are not the first ones to do work in the general direction of phishing attacks – please find a lot of related work from the area in chapter 3 – but within this thesis I tried to address the special aspect of bringing the technical security side and the user based interface side together and look more closely at their interplay within the scope of phishing attacks (see section 1.5). The essence of the research at hand can be defined using three terms: protection, detection and user intervention.

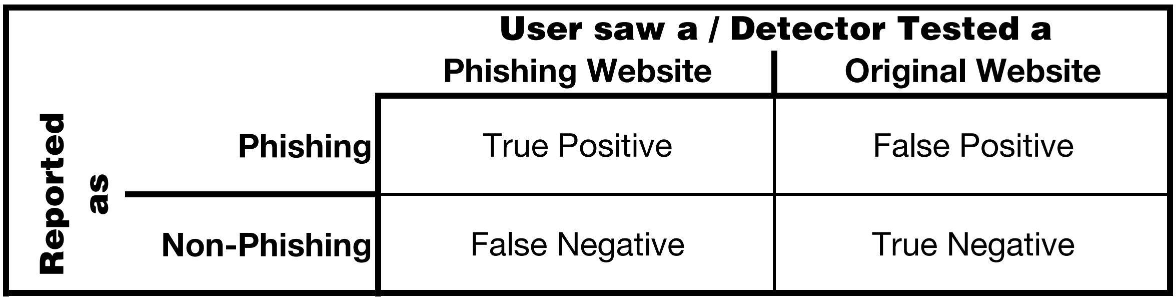
1.3 Protection: Detection plus Intervention

From the point of view of this thesis anti-phishing measures can only work well in case two different aspects come together to form a final **phishing protection** of the user. Hence we want to establish the following definition:

$$\textbf{detection} + \textbf{user intervention} = \textbf{protection}$$

I understand the process of phishing **detection** as a technically-originated search for existing attacks within a set of candidate websites. Usually a software detector will therefore take a website as an input and test whether it is a phishing website or not. The detection process ends with the output of the result.

User intervention (latin: intervenire$\widehat{=}$come between) is hence the follow-up step taking the result of the detector into account and using it for further actions towards the user. From simply blocking the network traffic associated with this website up to leaving the traffic untouched and displaying the result somewhere everything that follows between the detection and the user interaction with the website counts as user intervention.



		User saw a / Detector Tested a	
		Phishing Website	Original Website
Reported as	Phishing	True Positive	False Positive
	Non-Phishing	False Negative	True Negative

Figure 1.1: Matrix for true and false positives and negatives in the area of phishing detection.

Phishing **protection** finally stands for the fact that a phishing attack launched against a user is not only detected but that it is also guaranteed that the user is not actually harmed by the attack and finds her way successfully around it. Besides the optimum case of protection the process can also lead to other results (see section 1.4). I claim that without respecting both summands of the equation within the development of new concepts something important will be missing which can be exploited by an attacker to reach his goal.

A perfect detector that has no effect on the actual interaction would be worth nothing without a proper user intervention method and a well working user intervention method including a perfect warning dialog would not work in case no phishing website detection has been performed. Within this thesis I will have a look at the interplay between those components and show that they have a much stronger coupling than one might initially think.

1.4 Technical Terms of Detection

Throughout this thesis, especially when measuring phishing detector performance or the performance of users when classifying websites, a lot of different technical terms from signal detection theory will be used over and over again. As a point of reference I want to give a short introduction by exploring these properties. Whenever applicable I will try and give a practical example for the technical terms wherever used.

The four most important measurements for a detector are true positives, false positives, true negatives and false negatives (see figure 1.1 for an explaining matrix):

- Tasks that have been positively classified as an attack by the detector/user may either be:
    - **True Positives (TP):** This means the detector/user has correctly detected an attack. The number of TPs is high whenever security awareness has been raised in the right moment or the detector is performing well.

  - **False Positives (FP):** In such a case the detector/participant mistakenly thinks a non-malicious website denotes an attack. Detectors that are too sensitive and classify many websites as being phishing will have a high number of FPs. In case of a user intervention concept, one that frightens participants instead of properly raising security awareness in the right moment will have a high number of FPs.

- Cases where users/detectors do not detect any attack and assume the website is legitimate are counted as negatives:

  - **True Negatives (TN):** In this case true negatives are legitimate websites that are identified as such. A detector that has a low number of false positives will automatically yield a high number of TNs as original websites are correctly classified. In case a participant sees an original website a TN is achieved if the participant has justifiably no doubts about the originality of the website.
  - **False Negatives (FN):** This happens if attacks are not spotted, so whenever a participant believes an attacking task is a legitimate one. In case of false negatives attackers would have been able to trick the participant into an attack. For a detector these are phishing attacks that the detector missed during the classification process.

Summing this up, the goal of any work on anti-phishing will usually be to increase the number of true positives whilst keeping the number of false positives as low as possible. False negatives and true negatives are reverse values that can be calculated depending on the TPs and FPs (e.g. $100\% - \text{TP}\% = \text{FN}\%$).

Having those values assessed, two important other factors can be derived: Accuracy and Precision [298, 337]. **Accuracy** is the number of true positives and true negatives – the number of correct decisions – divided by the number of all results. **Precision** is the ratio of how many hits were correct, thus being the number of true positives divided by the sum of true and false positives. As these are only compound values I will usually refrain from reporting these numbers within the thesis and report the FP/FN/TP/TN.

In case of many detectors, threshold values are involved that can be adjusted to define the numbers of websites being classified within the four different categories mentioned above. To see how classification values depend on this classification **receiver operating characteristics (ROC)** is a well known method of displaying the dependencies between true and false positives concerning certain thresholds [174]. For a ROC curve the amount of true positives (y-axis) in dependence of the amount of false positives (x-axis) is plotted for different thresholds. The larger the area under this curve gets the better a given detector is. A diagonal line throughout the chart is equal to random guessing. A ROC curve has been used in the result section of subchapter 5.8 in figure 5.43.

Another type of evaluation that I used within this thesis is plotting the run of the curve of false negatives against the run of the curve of false positives for different thresholds on the x-axis. This kind of diagram makes it easy to find the thresholds for which the number of false positives is equal to the number of false negatives. This equilibrium value can be used to compare different detectors. An example for this kind of diagram can be found in the evaluation of the project in subchapter 5.8 in figure 5.42.

1.5 Main Contributions

Besides giving a thorough overview and analysis of the phishing problem and the related work about phishing detection and user intervention methodologies I present nine projects that have been carried out to find new methods for phishing detection and/or user intervention and advance the field in the direction of user-centered approaches to phishing protection.

The prototypes use a range of different possible input factors for the detectors to gather insights about which type of detection might be best suited for a detector with special regard to the human-centered properties of detection. One result of this thesis regarding these properties is that visual comparison is most promising from a user intervention point of view but has to be refined to result in better detection performance.

Different types of user intervention methods (indicators or warnings) are used within the projects and I even present a new general kind of warning dialog called "semi-blocking" warnings that can be used in special cases to block the users from performing insecure actions without immediately disturbing them in their current course of actions.

For detector and user intervention evaluation within the different projects a wealth of different evaluation techniques have been used, from classical lab and field evaluations to modern online evaluations. The thesis reports the different characteristics of those evaluation strategies.

The projects and the research framing this thesis finally leads to answers to 10 different research questions split up into five levels regarding phishing detection and user intervention. These levels are: definition, HCI, measurement, enhancement and reason (see chapter 4 for details). Besides answering those research questions (see chapter 6) the thesis also includes a more practical part offering guidelines to various aspects of detector and user intervention development (see chapter 7).

After looking at this interplay of detection and user intervention for phishing incidents in the browser I will also have a look at how the findings within this example field of socially engineered security threats could be possibly generalized.

1.6 Structure

In total this thesis consists of eight chapters organized in three parts. About all chapters will be concluded with a section of take home messages that are meant to recapture the essence of the respective chapter and its subchapters before moving on to the next one.

Chapter 2 The Act of Phishing: This chapter gives an introduction into the phishing problem and its characteristics without yet targeting research concerned with the problem. The chapter answers the question of what a phishing attack is and why a great need for counteracting phishing attacks exists. It provides an overview on the different kinds of phishing attacks and tells the history of phishing. In the end of the chapter a look at the design space of phishing attacks and current browser security indicators is taken.

Chapter 3 Related Work: In this chapter the scientific related work towards different parts of the problem space is covered. Starting with related work towards the phishing problem the chapter continues with an overview over problems with existing detection methods and user intervention methods. After a short interlude on phishing education new research concepts for detection and intervention are presented before reporting more general literature about user study methodology.

Chapter 4 Overview of Research Covered: After having laid out the foundations of this thesis, this chapter will set the work of this thesis apart from the research of related work and will introduce the different research questions that will be covered within this thesis.

Chapter 5 Nine Research Projects on Phishing and Usability: The by far longest chapter of this thesis contains the nine different projects that have been carried out, each one in its own subchapter. Each subchapter will start with an indicator of the research questions covered within the sections and will end with individual answers towards each of the tackled research questions.

Chapter 6 Aggregated Results and Derived Recommendations: This chapter will summarize the results towards all ten different research questions within the five levels. Afterwards a short discussion about the generalization of the results towards more general research areas will be given before presenting a final model summarizing the interplay between detection and user intervention.

Chapter 7 Recommendations and Guidelines: This chapter provides a more practical go on the results presented before by applying them to possible future detection and user intervention concepts, also discussing the question whether a web without phishing is at all possible and how the situation could change in the future. The chapter ends with evaluation recommendations presenting different options for user studies and giving recommendations when to use which of those options.

Chapter 8 Conclusions and Future Work: The final chapter concludes this thesis with a retrospection to the contents and findings and provides a brief future outlook on how future research should look like in general.

Take Home Messages

- **1.1 Usable Security:** Usable security brings together the often contradictory search for perfect security combined with usability for the users. Social engineered phishing attacks have an even more important property hence being an optimal research subject for usable security.

- **1.2 Problem Statement:** Phishing detectors are developed for several years now and they can yield promising but no perfect results. Since fully automatic detection is unrealistic, it is important to involve the user in the final decision that is actually not her primary goal. How can this problem be handled to make phishing protection more successful?

- **1.3 Protection: Detection plus Intervention:** Anti-Phishing is more than just technically trying to detect phishing websites and there is more to it than just displaying a nice looking warning using a good user intervention strategy. I argue that its only the combination of both that can be successful and hence define: detection + user intervention = protection.

- **1.4 Technical Terms of Detection:** When measuring detection success it is all about true positives, false positives, true negatives and false negatives. These values can be combined to accuracy and precision or with the help of different thresholds be plotted to ROC curves or other diagrams.

- **1.5 Main Contributions:** Throughout nine different research projects this thesis collects findings for 10 different research questions laid out in five levels. On this way a variety of different concepts, evaluations and parameters have been tested that allow me to draw final conclusions about how perfect protection should look like and how the interplay problem can be best assessed.

Chapter 2

The Act of Phishing

Phishing is perhaps one of the most pressing problems of usable security for today's average user. Each phishing attack has not only a technical but also a social component making it a perfect study subject for the research interleaved between HCI and security.

Millions of dollars each year are lost by people falling for phishing [190] and not only financial damage can be caused with credentials that have been obtained through phishing. Stealing credentials can enable adversary companies or governments to infiltrate sensitive infrastructures. In a recent example the complete network of the New York Times had been infiltrated with stolen credentials allowing the attackers to gather information and even manipulate the contents of one of the worlds most important news sources [231].

This chapter provides detailed insights on what a phishing attack is (section 2.1) and reports the most important reasons for counteracting phishing in section 2.2. Section 2.3 will provide a landscape of all important attack vectors[1] and explain which of those have been covered in the projects of this thesis and why. A brief history of phishing and a possible future outlook can be found in section 2.4 before section 2.5 will take a closer look at the current design space of phishing websites. Finally section 2.6 will briefly explain what possible indicators in one of today's browsers a user would have to identify phishing.

2.1 What is a Phishing Attack?

Literature and Internet are full of definitions for what "phishing" really is. Possible definitions range from very figurative descriptions to very broad, scientific and general definitions.

Techterms.com [282] an Internet dictionary addressing the general public defines phishing very figuratively whereas phishtank.com [233] a website dedicated to the collection of exist-

[1] An attack vector is a way that is used to bring an attack towards the user (e.g. email)

ing attacks already includes possible attack vectors and possible protection into their definition. A similar definition is given by a blog article on computerworld.com [147] (not listed above). In many definitions email is named as the only possible attack vector [251]. This misses other attack vectors like for example social networks. More universal definitions are given by the anti-phishing working group (APWG) [14] and scientific literature [221] (p. 2; not listed above).

> **Phishing 1** *Phishing is similar to fishing in a lake, but instead of trying to capture fish, phishers attempt to steal your personal information.*
>
> **– techterms.com [282] –**

> **Phishing 2** *Phishing is a fraudulent attempt, usually made through email, to steal your personal information. The best way to protect yourself from phishing is to learn how to recognize a phish.*
>
> **– phishtank.com [233] –**

> **Phishing 3** *Phishing is an email fraud method in which the perpetrator sends out legitimate-looking email in an attempt to gather personal and financial information from recipients. Typically, the messages appear to come from well known and trustworthy web sites.*
>
> **– Rouse, techtarget.com [251] –**

> **Phishing 4** *Phishing is a form of online identity theft that employs both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials.*
>
> **– Anti-Phishing Working Group APWG [14] –**

Perhaps one of the most valid and important definitions has been given by Steven Myers [137]:

> **Phishing 5** *A form of social engineering in which an attacker, also known as a phisher, attempts to fraudulently retrieve legitimate users' confidential or sensitive credentials by mimicking electronic communications from a trustworthy or public organization in an automated fashion. Such communications are most frequently done through emails that direct users to fraudulent websites that in turn collect the credentials in question.*
>
> **– Steven Myers [137] –**

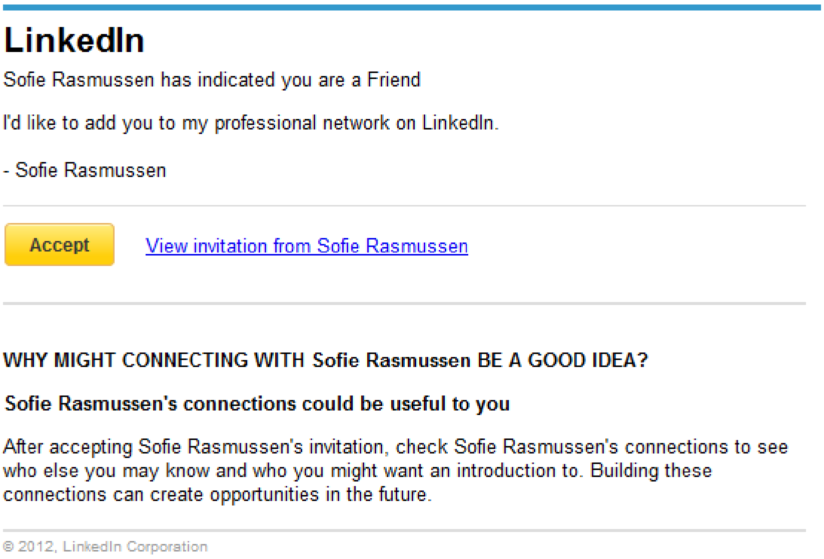
As the above definitions illustrate the process of phishing involves several steps each of which also gives the potential of counteracting the attacks. This thesis heavily focuses on the fraudulent websites used for collection and possible ways to counteract phishing there. This results from the fact that most of the user interaction during the phishing process happens there and hence this process of interaction offers the largest possibilities for successful intervention. Taking this into account the following definition might be suitable for Phishing in regards to this thesis.

> **Phishing** *Phishing is an act of trying to get hold of sensitive data and valid user credentials by luring users into entering these on legitimate looking websites. Attackers try to maximize the number of successfully fooled users whilst minimizing the chance of their attack being detected and removed by an authority.*

As outlined in section 2.3 this protection of the user on websites can only be accomplished when looking at the complete process of phishing from the perspectives of each individual stakeholder.

From a user's perspective a phishing attack might look like a simple visit on one of her favorite websites that shows a slightly weird behavior. Perfect attacks will completely go unnoticed and could happen as follows:

Alice is looking at her inbox and looks at an email she just received from her favorite business networking platform indicating that a new business contact "Sofie Rasmussen" seems to request her profile (see figure 2.1). She wants to accept the invitation and presses the button shown in the email. Her browser opens and the LinkedIn login page appears. She enters her credentials and submits the form. It somehow seems that the credentials were not entered correctly as the login page reappears. Alice enters her credentials again and is successfully logged on to her account. She cannot find an invitation of "Sofie Rasmussen" and logs out again. Probably the contact has revoked the request.



LinkedIn

Sofie Rasmussen has indicated you are a Friend

I'd like to add you to my professional network on LinkedIn.

- Sofie Rasmussen

WHY MIGHT CONNECTING WITH Sofie Rasmussen BE A GOOD IDEA?

Sofie Rasmussen's connections could be useful to you

After accepting Sofie Rasmussen's invitation, check Sofie Rasmussen's connections to see who else you may know and who you might want an introduction to. Building these connections can create opportunities in the future.

© 2012, LinkedIn Corporation

Figure 2.1: Screenshot of a real phishing email received in July 2012. This email passed our university's spam and email-filters and reached its intended recipient.

In the above example Alice just fell for a well made phishing attack and she isn't even aware of it yet. Her first login to LinkedIn was not on the real LinkedIn website but instead she had entered her credentials at a phishing website mimicking the real LinkedIn login portal. After the first submission the phishers collected her data and simply redirected her to the real login page of LinkedIn which made the second login attempt successful. "Sofie Rasmussen" did not show up as she never tried to get in touch with Alice.

The attacker's perspective of a phishing attack is completely different. He has to get hold of email addresses of possible subjects, has to set up a convincing website that is able to get hold of the credentials and finally has to make use of the credentials gathered. A detailed report on how phishers compromise hosts is given by Watson et al. [292]. They set up a honeypot[2] server that reported all steps done by attackers to compromise the system and set up a phishing attack.

The above example is only one of many different attacks that could happen and the state of the art phishing attacks will be more closely explained in the forthcoming chapters.

2.2 The Need to Counteract

Many important authorities have made up their mind what the biggest problems about phishing are and why the problem needs to be taken care of. However, the possible advice given in these documents most often does not reach the intended recipient.

[2] A honeypot is a trap setup to deliberately catch malicious attacks and learn from them [302].

The National Consumers League (NCL) as Americas oldest consumer organization [214] has published a 66 page document titled “A Call for Action” [215] in 2006 to show the importance of anti-phishing countermeasures. The document summarizes the results of a special anti-phishing retreat held by major governmental and commercial institutions. They report an increase in Phishing attacks from 176 in January 2004 to 4,367 in October 2005. Data from phishtank.com [234] (see section 5.1.2) shows a total number of more than 25,000 attacks in January 2013. That is 140 times more attacks in less than 10 years. The NCL also found signs of declining trust of users to the Internet.

Attackers gather in groups like the “rock-phish group” for example to develop ever more sophisticated attack types. Moore and Clayton [203] discovered some of their methods using relaying of domains and “fast-flux” networks (see section 2.3). Their websites stay up longer (avg. 454 hours) than the average 58 hours of a standard attack. However, this time frame is still large enough to collect data of users. Eugene Kaspersky speaks of an explosion of cybercrime and malware [170] and suggest drastic measurements to protect users. They should have a personal online ID that they need for Internet access on any website. Although a measure like this could help to find attackers more easily it would still create new possibilities for fraud and more importantly privacy issues.

Richard Clayton [52] thinks that the biggest issue with this kind of fraud is the more or less direct transfer of real world authentication protocols to the virtual world where it is much easier for an attacker to disguise as a trusted party.

The NCL [215] also gives some recommendations for possible actions to be taken. The first of those is user education a highly controversial topic that is more closely examined in section 3.4. Besides this a consumer experience should be “secure by design”. This principle that can only be fulfilled to a limited extent with the current Internet architecture. Although most banks and online retailers use secure channels for transmitting consumer data they still have no control at all over users falling for an impersonation attack on a server outside of their companies control radius. Besides this, the NCL suggests black- and whitelists to get hold of the problem. However, these state of the art measures are not enough to stop more phishing attacks being launched (see section 3.2). As the NCL outlines, a better understanding of users about phishing attacks would be a step in the right direction but this seems to be impossible without better technical detection methods that are superior to black- or whitelists (see section 3.2.1). Such concepts and the way the can be developed are in the heart of this thesis. Phishers learn from their victims and as they do research should look at users and attackers to create working counteractive measures.

As the phishers successfully form groups to be more effective counteracting agencies would also need to. In 2008 Moore and Clayton [205] proved that with better cooperation between “take-down”-agencies many more attacks could have been taken offline earlier. However, world-wide cooperation of law enforcement or private “take-down”-companies is hard to establish, similar as changing protocols of huge systems like the Internet that is deployed globally (see section 2.3.1 for the steps of a phishing attack and the counteracting possibilities). With a definitive need to take action I try to counteract attacks not at the technical bottom layers (like protocols). Instead the concepts presented later on try to detect attacks and protect the user right before the impact of the attack.

Florênico and Herley [93] argue that the problem for the end user is smaller than perceived by the general public. Illegally made online banking transfers usually can be reverted and the customers in fact have no financial loss in the end. They argue that the mules used by the phishers to launder the money are the ones that actually loose money as they end up with a negative account balance after a transaction has been revoked. For some financial transactions this might be true but the core problem still persists: sensitive data of individuals is stolen and misused. The misuse with this data not necessary needs to be of financial kind and even if the victim is reimbursed the crime committed should still be stopped.

2.3 Phishing Attack Overview

Defining the scope or the types of a phishing attacks is even harder than finding a legitimate definition for the term of phishing. What is part of the attack? Is it only the action taken inside the browser or is the whole process from planning to the final monetization. The first subsection (2.3.1) has a look at different birds eye views on phishing attacks and defines the exact area of an attack that it is part of this thesis. Subsection 2.3.2 first gives a short explanation of common attacks outside of the scope of this work and explains how these are related to the overall problem. Finally in subsection 2.3.3 the current attacks that are important for the research in this thesis are explained.

2.3.1 The Lifecycle of a Phishing Attack

Several books and papers have looked at phishing from a birds eye view at different heights. Some papers describe the linear process of phishing in several steps. According to Steven Myers [212] a phishing attack in general consists of “The Lure” (e.g. email spamming), “The Hook” (e.g. a phishing website) and “The Catch/Kill” (e.g. identity theft using the credentials”). Aaron Emigh [81] uses a more detailed eight-step-model. Step 0: a preparation phase (e.g. for registering a domain) is followed by sending out a malicious message (step 1) that is somehow responded to by the user (step 2). This is followed by the prompt to provide confidential information (step 3) and the user’s answer to this prompt (step 4). This confidential information is then transmitted back to the attacker (step 5) and used to impersonate the person (step 6) which can then be followed by means to engage in further fraud (e.g. monetizing the data) (step 7). The National Consumers League sees Phishing as a six step process, consisting of “Plan Attack”, “Launch Attack”, “Gather Data”, “Research How To Use Data”, “Attempt Crime” and “Launder Proceeds”. Figure 2.2 shows the different life cycles side-by-side. Besides the fact that they differ in granularity the specified end of the life cycle differs a little. While Myers sees the fact that data was obtained more or less as the end of the lifecycle, Emingh expected further fraud to be going on with the data and the

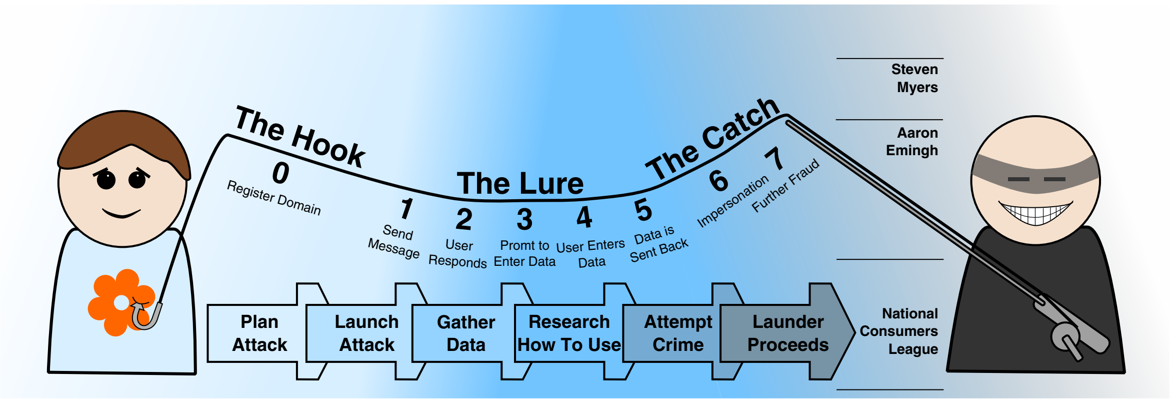


Figure 2.2: An illustration of the lifecycle of a phishing attack as described by Myers [212], Emingh [81] and the National Consumers League [215]

National Consumers League even takes the "laundering" process of the gained goods (e.g. money, credentials) into account.

It is important to mention that at every step of this lifecycle an intervention to prevent phishing would probably be possible. The aforementioned references [81, 212, 215] give some examples for that. In case it would be possible to guarantee that malicious people cannot get hold of a domain or website in the first place, this would stop phishing as well as if it would be impossible to use the gained confidential information in the end. In this thesis my focus is mainly on the steps two to five (according to Emingh [81]) which is the phase where the user is actually inputting her secrets. These steps seem to be the most promising to build upon, as protecting the user there, can be accomplished without the need to change well established structures like Internet protocols or law enforcement borders.

The exact possibilities for the execution of the different attack steps are endless and this makes it possible to unfold the phishing life-cycle into what could be named as a "map of phishing". Looking at the different stakeholders ("bad guys", "good guys" and "consumers") Jeffrey Friedberg [96] created a graphic titled "Internet Fraud Battlefield" (see figure 2.3 or A.1 for a larger version). Ollmann [221] focused more on different ways and purposes of phishing attacks in an overlook on "the methods used in phishing" (see figure 2.4). I tried to compile the most important facts for this thesis into a flow chart: the "map of phishing" (see figure 2.5). It shows the different attack vectors, the most common phishing types and scenarios, the possible delivery channels and the different ways of data collection using either malware, malicious websites or the very uncommon dial-in attempts. All these actions finally aim for different purposes. All these attacks are highly interactive such that a malicious website can be used for multiple purposes (e.g. to gather information and install malware using an exploit) [62]. Markus Jakobsson [135] also presents an approach for creating graph-based models of individual phishing attacks. These can afterwards be used for different means (e.g. for calculating the probability for a successful attack).

Walking through the "map of phishing" one of the more important attack vectors for phishing at the moment definitively are email messages. With all the possibilities that spam emails

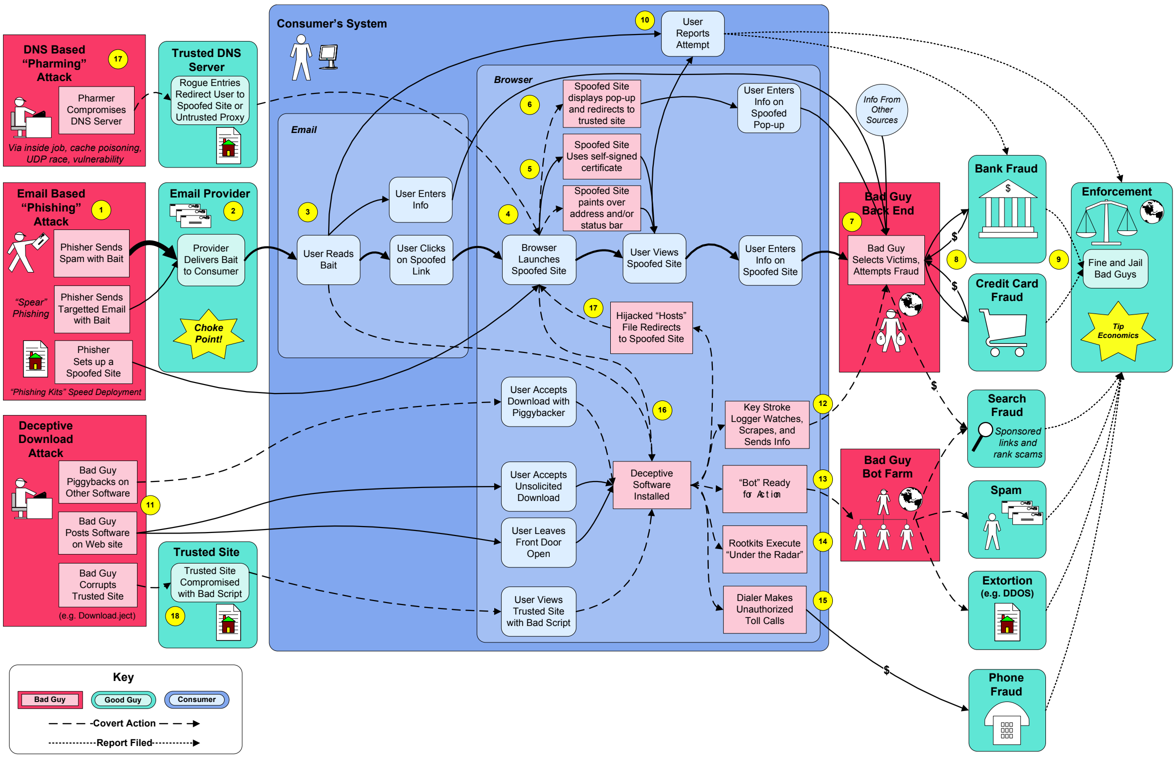

DNS Based "Pharming" Attack
17
Pharmer Compromises DNS Server
Via inside job, cache poisoning, UDP race, vulnerability
Trusted DNS Server
Rogue Entries Redirect User to Spoofed Site or Untrusted Proxy
Email Based "Phishing" Attack
1
Phisher Sends Spam with Bait
"Spear" Phishing
Phisher Sends Targetted Email with Bait
Phisher Sets up a Spoofed Site
"Phishing Kits" Speed Deployment
Email Provider
2
Provider Delivers Bait to Consumer
Choke Point!
Deceptive Download Attack
Bad Guy Piggybacks on Other Software
11
Bad Guy Posts Software on Web site
Bad Guy Corrupts Trusted Site
(e.g. Download.ject)
Trusted Site
Trusted Site Compromised with Bad Script
18
Consumer's System
Email
Browser
10
User Reports Attempt
3
User Reads Bait
User Enters Info
User Clicks on Spoofed Link
4
Browser Launches Spoofed Site
6
Spoofed Site displays pop-up and redirects to trusted site
5
Spoofed Site Uses self-signed certificate
Spoofed Site paints over address and/or status bar
User Enters Info on Spoofed Pop-up
User Views Spoofed Site
User Enters Info on Spoofed Site
17
Hijacked "Hosts" File Redirects to Spoofed Site
User Accepts Download with Piggybacker
16
User Accepts Unsolicited Download
User Leaves Front Door Open
User Views Trusted Site with Bad Script
Deceptive Software Installed
12
Key Stroke Logger Watches, Scrapes, and Sends Info
13
"Bot" Ready for Action
14
Rootkits Execute "Under the Radar"
15
Dialer Makes Unauthorized Toll Calls
Info From Other Sources
Bad Guy Back End
7
Bad Guy Selects Victims, Attempts Fraud
8
$
$
$
$
Bad Guy Bot Farm
Bank Fraud
Credit Card Fraud
Search Fraud
Sponsored links and rank scams
Spam
Extortion (e.g. DDOS)
Phone Fraud
9
Enforcement
Fine and Jail Bad Guys
Tip Economics
Key
Bad Guy
Good Guy
Consumer
Covert Action
Report Filed

Figure 2.3: The "Internet Fraud Battlefield" gives an overview over different attack vectors, consumer vulnerabilities and the possibilities for final fraud [96] (see figure A.1 in the appendix for a larger version).

in general offer, it is easy to reach a large audience in a matter of seconds with a small amount of effort. Spam emails and phishing attacks are heavily correlated [208] and spam is not only sent out together with the launch of an attack, instead more emails are sent out as long as the attacks stay online. About 21% of the spam usually arrives a day before the attack is detected, 46% within a day of the detection and 33%, more than a day after the detection. Social networks and social media as a target of phishing attacks have been growing immensely [179] but they are also used as attack vectors [178]. Other attack vectors, like banner advertisements or instant messaging are used far less. Besides actively contacting the victim it remains also possible to create own malicious content and have it indexed by search engines to wait for users to visit the malicious website on their own [81].

Regarding the different types of phishing that can occur, the number of different terms steadily rises. Besides the "standard" phishing attack, spear phishing is a special form of phishing that uses specific knowledge about the victims for better impersonation [220]. As an example emails are crafted that look as if they come from the victim's employer. Whaling, is a special form of these kind of attacks that focuses on important people in huge companies or government representatives [221]. Clone phishing uses original communication contents (like a standard invoice of company or a boarding pass email) and clones the content of such an email by just changing the contained links or the attachments [19].

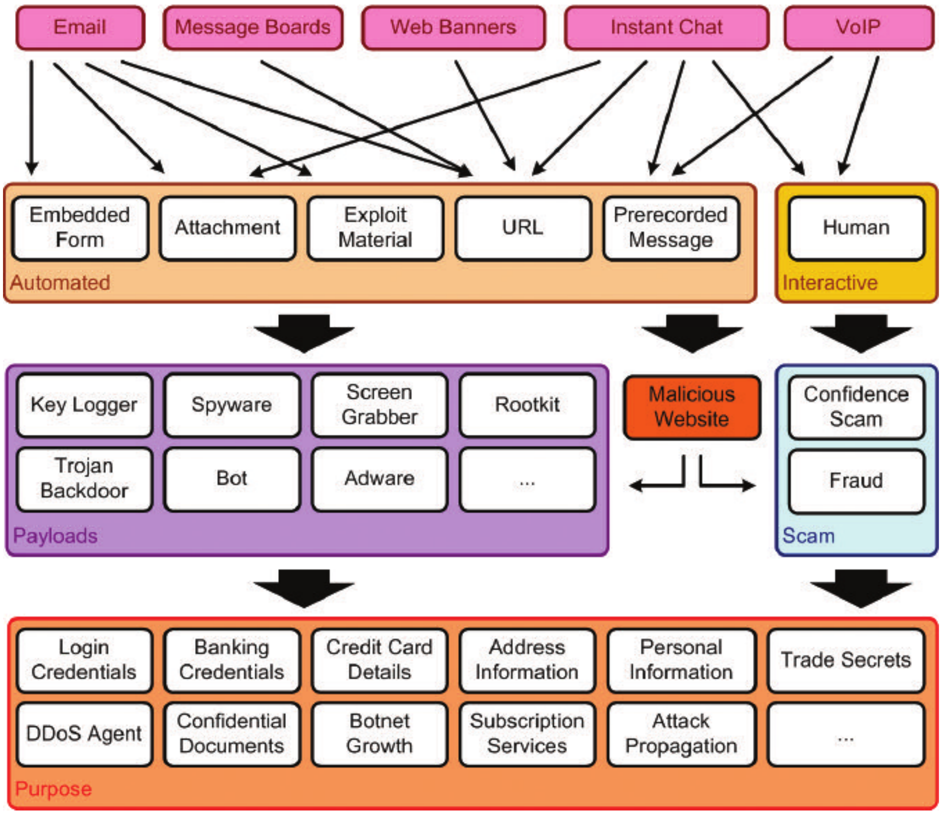


Figure 2.4: The different methods used in phishing as identified by Gunter Ollmann [221]

Several other types of Internet attacks are closely related to phishing but do not match the definition of phishing used in this work. Scams for example are sent out to people with the target of tricking them into participating in some form of activity that will be worthwhile for them. At some point during the scam the victim is asked to execute a specific action (e.g. paying a fee) that seems to be worth it. For this kind of attack usually no website or other technical means are needed to make it different from a classic phishing attack. Scams also have other limitations: they are specially crafted to not convince as many people as possible but only convince people that are also likely to fall for the later action [124]. Drive-by-pharming is another type of attack carried out by malicious emails to modify the users home router settings for malicious use [274]. In this case no confidential information is collected which differs a lot from phishing attacks.

No matter from which attack vector an attack originates from, there are a lot of different scenarios that are used to convince the user to react to the stimulus. Typical email topics can be the need for a security upgrade of the users account, incomplete account information, a special financial incentive or other account problems [212]. Besides, the impersonation of well known content of company emails is increasingly often used. This coined the above mentioned term of “clone phishing” attacks. Another well-known attack type to find user credentials is the so-called “man-in-the-middle attack” [85]. In this case an attacker sits

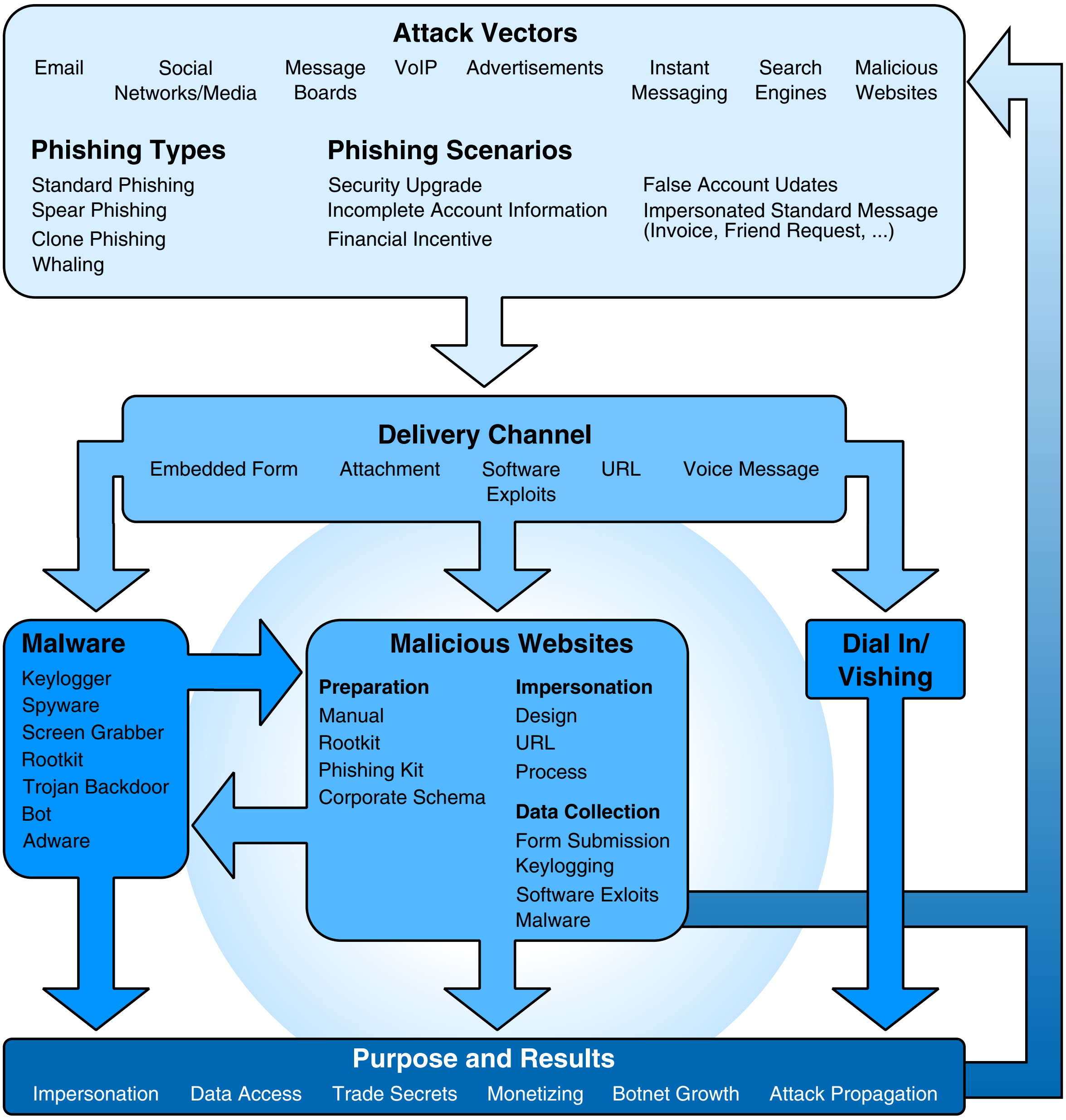


Figure 2.5: The "Map Of Phishing": An Overview of attack vectors to purposes for phishing attacks. Some items are compiled from other literature [96, 221]. The blue highlighted area shows the central aspects of this thesis.

between the user's computer and the companies server in the network (either by introducing itself as a proxy-server or by other means). All traffic that passes this network entity can now be read by the attacker extracting confidential information without the user noticing this. For encrypted connections this is nearly impossible as the attacker either would need to know the secret key of at least one party or would have to set up an encrypted relay attack, which could be noticed by the user. Using eavesdropping it is hence possible to overcome even two-factor authentication[3] [149]. Since this type of attack does not have a social engineering component, it cannot be labeled as a phishing attack. To impersonate a third party and acquire user credentials a connection to the original server is not most of the time not required.

In conjunction with such attacks the term "spoofing" is often heard. Originating from network security this term usually stands for masquerading as another network entity. "Email spoofing" denotes the fact of sending emails that virtually seem to be sent from a trustworthy source. "Web spoofing" is hence more or less a synonym for a phishing attack. And lastly "IP-spoofing" allows to send packages into the Internet that seem to come from a different source than they really are [115]. This can hardly be used for phishing attacks as answers to such packets do not reach the sender of the spoofed network package.

Depending on the attack vector different delivery channels can be used to redirect the user to a point of a data extraction. In the most basic case this can be an email containing an HTML-form for data entry. Since standard email clients usually warn extensively about these types of forms and since they are never used in trustworthy company communication they are seldom used. Software exploits of mailreaders or Internet browsers can be used to directly execute malicious code on older systems but users with up-to-date systems are most of the time protected of those attacks. Usually a somehow masqueraded URL (or link) is one of the major entry points for phishing websites whereas malware attacks usually come as attachments. Some other messages try to make the victim call a certain automated phone service that will then ask for confidential details in an automated manner. This dial-in phishing over VoIP is often called "Vishing" [30].

A joint report of the US department of homeland security, SRI international identity theft council and the APWG summerizes all these attacks under the overall term "crimeware" [80]. Concerning malware, any installed software can usually gain full access over the infected machine. Malware does hence not only pose a risk to confidential credentials of the user but to much more. For a user and other software it is hardly impossible to detect ongoing attacks, once successfully installed on the victim's machine. For these kinds of attacks virus scanners and other detectors are the more important means for detection; HCI is less important for their development process. This is the reason why this thesis mostly focuses on socially engineered phishing websites that try to gather confidential information. In addition to that phishing itself is more often conducted using such websites than through

[3] Multi-Factor authentication uses multiple authentication factors for one authentication attempt [306]. A password would be considered only as one factor. Adding a physical token to the authentication process (e.g. a smartcard) would then be two-factor authentication. German banks use one time passwords as a second factor for authenticating bank transactions.

malware. On a malicious website the attacker can ask directly for the data he is interested in whereas using malicious software he has to extract the data needed from the vast amount of data that can be gathered on the victim's computer.

Looking at the Google Transparency Report [112] showing numbers of malware and phishing websites on their indexes phishing is on the rise although there are about 2.4 times as many malware sites as phishing websites registered – 149k phishing websites vs. 360k malware websites on 2013/07/14.

Moore and Clayton [207] offer another interesting perspective on how attack targets in terms of Internet hosts[4] can be selected by phishers. Nearly 20 percent of all hosts that are infected with malicious websites are found by the attackers using search engines. For example, searching for the version number of a blog software that is known to have a vulnerability makes it possible to find working websites for attacks in no time.

In the following sections, first of all the types of concrete attacks that are out of focus of this thesis will be briefly explained followed by information on typical phishing attacks that are covered by the research in this thesis. For further reading a master thesis by Stan Hegt [120] gives more detailed information about the different attack types of phishing attacks.

2.3.2 Attacks out of Scope

Concerning malware-based attacks a variety of possibilities exist to inject malicious code into the victim's computer. Provos et al. [240] analyzed the most important mechanisms. Missing web server security can be used for an attacker to introduce malicious content to a website as well as possibilities to put user contributed content onto a website that does not do input sanitization. Adverts that are propagated to many websites could contain malware as well as third-party widgets on a website.

These content injection attacks can be used for both: malware downloads or redirects to malicious websites. Looking at possible types of malware **keyloggers**[5] or **screengrabbing**[6] software can be used to record I/O-channels of the victim's computer. Later manual or automatic analysis can yield passwords and usernames that were entered on different websites [81]. **Session Hijackers** allow attackers to read out the session-ID of a user on a specific website after the victim has logged-in. Such a session-ID (usually stored in a cookie) identifies all subsequently made requests towards the server as being originated by the same user. Setting a valid session-ID on a different computer allows the hacker to impersonate the currently logged in user from another machine [81]. **Web Trojans** are software versions of phishing attacks asking for credentials in specifically designed popups [81]. **Spyware** can be seen as an overarching term for software sending specific user data back to the servers of

[4] A Host is a computer connected to a network – in this case the Internet [303].

[5] A software running in the background of a computer capturing the keyboard input.

[6] A software running on the computer capturing the screen contents.

an attacker [236]. **Adware** usually is a less intrusive software that tries to generated revenue for its use by displaying ads [236]. A **rootkit** is a software program that tries to open up super user privileges for a third party without the original user noticing this [188]. This can be often accomplished using a specific type of malicious bot[7].

As mentioned before this work does not focus on the attack vector that was used to redirect the user to the phishing attack – in most cases an email. Nevertheless it is is also possible to detect possible attacks already at this level. Drake et al. [75] describe the typical contents of such an email and the tricks being used in there: impersonation of a company's logo or style; the presence of a different reply-address; a plausible premise for the email and the inquiry to quickly respond together with security promises are only some of the many possible indicators for a phishing email. Although all indicators seem very sound it is hard to detect this kind of semantic information in an automated manner.

2.3.3 Attacks in Scope: Impersonation

Whenever a malicious website is set up, the attackers can choose to impersonate a certain brand for the credentials that they want to collect. To make their impersonation more trustworthy they have the possibility to use three different styles of impersonation: **design impersonation**, **URL impersonation** and **process impersonation**.

- **Design Impersonation** Cloning the design of a website lets the malicious site look more closely like the original. In general, this can simply be achieved by copying the HTML code and all images of the original website to a different host.
- **URL Impersonation** In most cases it is not possible for phishers to compromise the actual host that should be impersonated and if so they will eventually have direct access to all user credentials without setting up a malicious website. This means, phishing websites have to be set up on other servers with a different URL. However, the attackers use a lot of different attacks to make a URL look more convincing and closer to the original.
- **Process Impersonation** Besides the design of a website the way a website behaves and its interaction possibilities can also be cloned to some extent. Many websites contain non-static content that is generated based on certain personalized and non-personalized aspects (e.g. news feeds, recent items, friend feeds). Impersonating this content based on user preferences is hard for an attacker that does not know that data. A standard login form may already raise the user's suspicion if it asks for more data than the original website usually does.

To be able to speed up the setup process of attacks and to ease the design impersonation phishers make use of different software tools [212]. **Rootkits** as mentioned before automate

[7] A bot is a small software to perform tasks on computer and interact with another network entity [51].

the process of hacking into a computer and maintaining the possibility for super user access. **Phishing kits** automate the setup procedure of a phishing website sometimes even including the design impersonation for an existing website. For other attacks phishers also use so called **"corporate schemas"** that contain the standard look and feel for a specific brand and can be quickly used for the setup of a phishing website. Phishing kits are in widespread use and are even used to defraud other attackers. Some of those kits collect the credentials not for the phisher that used the kit but instead for the original creator [57].

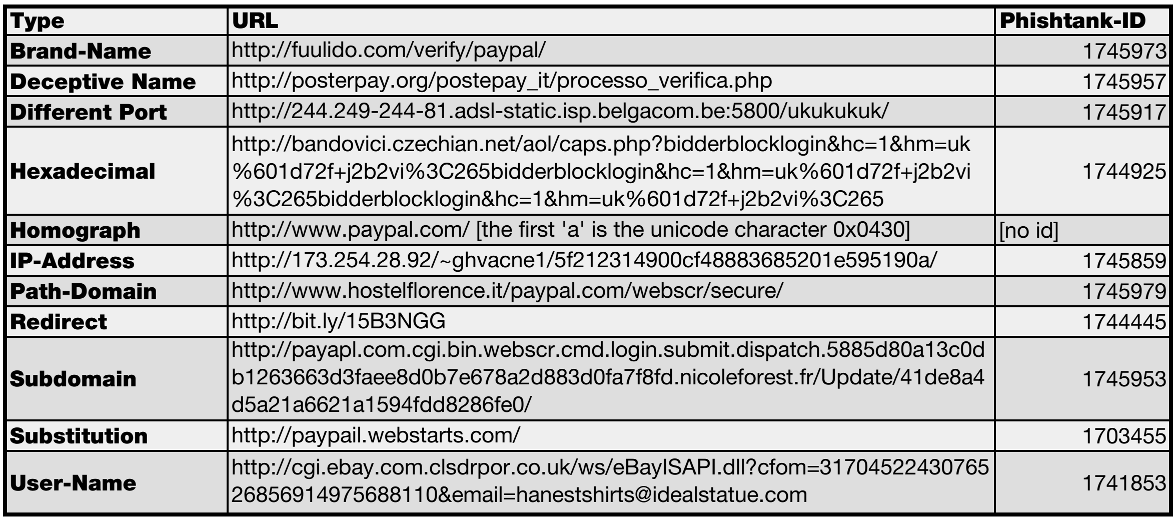
URL Attacks

Making the user believe that he visits a correct URL already starts with the email that contains the link. Different techniques are used to display a legitimate looking URL such that the user clicks on that link [189]: Using HTML-mails the link-text can differ from the actual link destination and JavaScript methods can be used to obfuscate the final URL even more. In some cases IP-addresses instead of URLs or the @-symbol are used to confuse the user when looking at the URL. Everything before the @-sign of a URL is used as the username of the connection[8]. Such a username can also have the name of the original URL of the website that is impersonated. Other options are to use hexadecimal characters inside the URL or a redirection service that is more trustworthy.

In other cases similar sounding domains are registered by the attacker using stolen credit card information or dubious hosting providers. Substituting similar looking characters (e.g. an "i" for an "l" in "paypai.com") makes it also possible to generate different but similar looking domain names. A special form of this is the homograph attack [103, 115]: it makes use of unicode characters that are visually very close. Domains with those characters can be registered as well, and hence look visually just about the same as the original URL. Other authors refer to it as the IRI-attack (Internationalized Resource Identifier) [100].

In other cases the original URL is enclosed in the subdomains of the website or in the path portion of the URL. For faking subdomains the attackers must have access to any domain to be able to control possible subdomains. Some free hosters also offer to choose an arbitrary subdomain for a blog (e.g. "xyz.wordpress.com"). The path portion can be controlled on any webspace by simply creating folders with the desired name of a target website. Table 2.1 shows a list of the different URL attacks and an example of a real phishing website with that type of domain found at phishtank.com. Many kinds of URL attacks will play an important role in the project presented in subchapter 5.4.

[8] In some cases a website needs a username and password transmitted to the web server. This can be encoded as an optional parameter in front of the URL using the @-sign.



Type	URL	Phishtank-ID
Brand-Name	http://fuulido.com/verify/paypal/	1745973
Deceptive Name	http://posterpay.org/postepay_it/processo_verifica.php	1745957
Different Port	http://244.249-244-81.adsl-static.isp.belgacom.be:5800/ukukukuk/	1745917
Hexadecimal	http://bandovici.czechian.net/aol/caps.php?bidderblocklogin&hc=1&hm=uk%601d72f+j2b2vi%3C265bidderblocklogin&hc=1&hm=uk%601d72f+j2b2vi%3C265bidderblocklogin&hc=1&hm=uk%601d72f+j2b2vi%3C265	1744925
Homograph	http://www.paypal.com/ [the first 'a' is the unicode character 0x0430]	[no id]
IP-Address	http://173.254.28.92/~ghvacne1/5f212314900cf48883685201e595190a/	1745859
Path-Domain	http://www.hostelflorence.it/paypal.com/webscr/secure/	1745979
Redirect	http://bit.ly/15B3NGG	1744445
Subdomain	http://payapl.com.cgi.bin.webscr.cmd.login.submit.dispatch.5885d80a13c0db1263663d3faee8d0b7e678a2d883d0fa7f8fd.nicoleforest.fr/Update/41de8a4d5a21a6621a1594fdd8286fe0/	1745953
Substitution	http://paypail.webstarts.com/	1703455
User-Name	http://cgi.ebay.com.clsdrpor.co.uk/ws/eBayISAPI.dll?cfom=31704522430765268569149756881 10&email=hanestshirts@idealstatue.com	1741853

Table 2.1: Examples for the different types of phishing URLs taken from original phishing URLs found by phishtank.com (except for the homograph attack).

2.4 A Brief History of Phishing and a Possible Future Outlook

Phishing as defined in this thesis originated in the 1990s. America Online (or AOL) was an Online Service that was popular back then. For a reasonable amount of time it allowed to create accounts with any credit card number that passed a standard validation check. Many hackers used this to create AOL accounts for free. These accounts lasted until AOL first tried to bill the respective credit card detecting that it did not really exist. When AOL closed this security hole – by immediately checking the credit cards at registration time – the hackers switched to what was then known as phishing. They contacted arbitrary AOL customers and posed as employees of AOL asking the customer's password for security purposes. Using these passwords they were able to log in to the user's AOL accounts and could steal more confidential data there [212].

2.4.1 The Term "Phishing"

Most literature about phishing refers the term of phishing back to a Usenet newsgroup post of 1996 in the hacker newsgroup "alt.2600" [212, 221].

A full reprint of this Usenet post and its replies can be found in the appendix in figures A.2 and A.3. Reading the whole post one can even understand a little bit of the phishing history and how people used to make use of generated credit card numbers.

> **AOL for free?** *It used to be that you could make a fake account on AOL so long as you had a credit card generator. However, AOL became smart. Now they verify every card with a bank after it is typed in. Does anyone know of a way to get an account other than phishing?*
>
> **– mk590, alt.2600, 28 Jan 1996 [200] –**

In fact, in the above post the term is already used as if it was common language. No further explanations come with it and the people replying to the post did not ask for any. Looking a little further back in the same newsgroup I stumbled along a post that from about one year earlier. A user called "Rick Buford" replied to a post from a user that had problems creating a new AOL account.

> **AOL Cert: lugs-lousy** *not anymore..just got a new one today....*
> *So Long and Thanks for all the Phish....*
>
> **– Rick Buford, alt.2600, 5 Feb 1995 [278] –**

This clearly is a reference to Douglas Adams fourth book of the "The Hitchhiker's Guide to the Galaxy"-series called "So Long, and Thanks for All the Fish [4]". But the misspelled "phish" could be another much earlier reference to the term of phishing in this case. As Ollmann [221] states, hacked accounts where actually called "phish".

Another clear evidence for the term of "fishing" being synonym for password stealing is a scientific paper from 1990 (five years earlier) [118]. This paper describes what could be also called one of the first malwares. A small computer program hooking into the I/O of a computer waiting for the word "Logon:" to appear and then record the username and password of a system user. This tool was called "FISHES".

But where does the "ph" come from? Some sources claim that this is borrowed from the older act of "phone phreaking" [165] where the telephone communications standards are misused to make unauthorized calls.

The Evolution of Phishing

Phishing evolved quickly [221] and got more professional over the years making use of new technologies like keyloggers and later on screengrabbers. New terms like "spear phishing" and "pharming" – redirecting users to phishing websites by technical manipulation of the network – evolved. The timeline in figure 2.6 provides and overview how phishing evolved over the years.

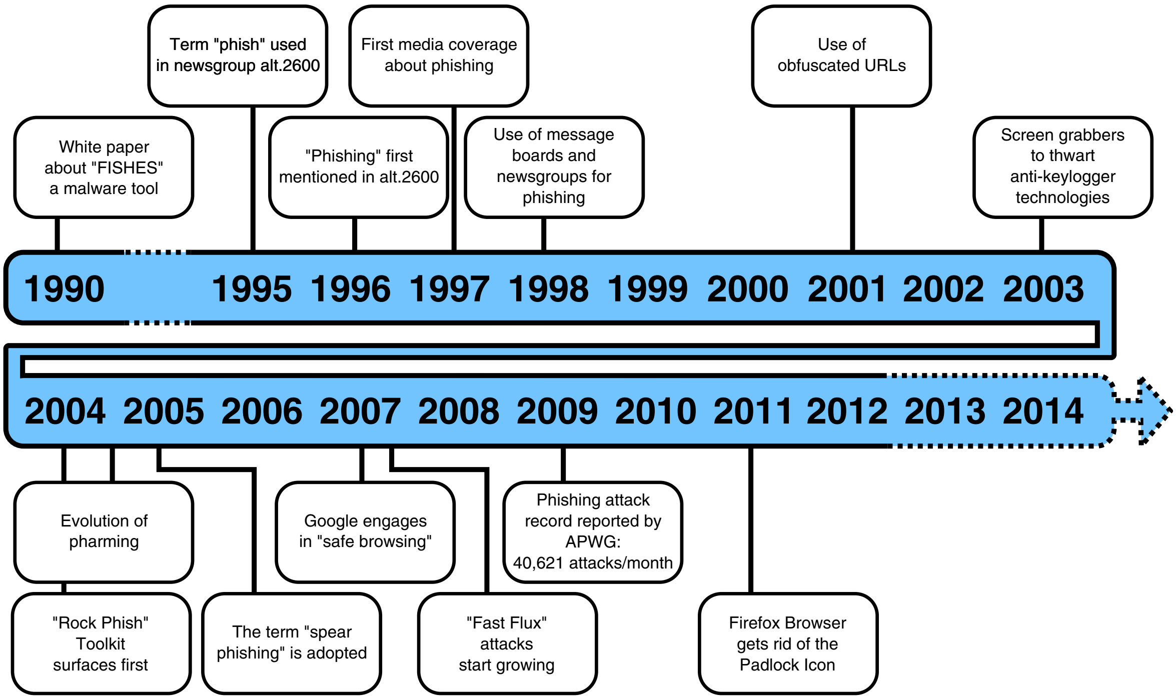

Term "phish" used in newsgroup alt.2600
First media coverage about phishing
Use of obfuscated URLs
White paper about "FISHES" a malware tool
"Phishing" first mentioned in alt.2600
Use of message boards and newsgroups for phishing
Screen grabbers to thwart anti-keylogger technologies
1990
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
Evolution of pharming
Google engages in "safe browsing"
Phishing attack record reported by APWG: 40,621 attacks/month
"Rock Phish" Toolkit surfaces first
The term "spear phishing" is adopted
"Fast Flux" attacks start growing
Firefox Browser gets rid of the Padlock Icon

Figure 2.6: A timeline of the history of phishing (based on [221] with additional sources [15, 165, 187, 200, 251, 272, 278, 286])

As phishing got increasingly popular over the years the fight against it slowly began. First the attacked companies started the fight against phishing on their own [241]. In 2007, huge companies like Google started initiatives to protect the users [187]. A central blacklist server for the Firefox and the Google Chrome browser was set up. Early research on why people fall for phishing started after the year 2000. This kind of research will be more closely explained in section 3.1.

Over the years phishers have learned to adapt to appearing countermeasures and have perfected their attacks using techniques like fast-flux networks, distributed phishing attacks and generated domain names, common countermeasures like blacklisting can be easily avoided (all to be explained later in section 3.2). Using the aforementioned phishing kits and corporate schemes incredibly convincing fake websites can be created.

Although the amount of phishing is not dropping, it does not seem to be an overly lucrative business. Extracting money from accounts may be hard [93] and the more people engage in phishing the less profitable it gets as the number of victims is limited and even drops as more people attempt the crime [125]. But it's not all about money and as more information about companies and ourselves is going online everyday the reasons for stealing confidential information rise, too.

But how about the future? Will phishing be able to find ever new ways of countering the methods that are developed to counteract it? Figure 2.7 shows how better detection drives the phishers into creating websites that cannot be automatically detected. Within this thesis

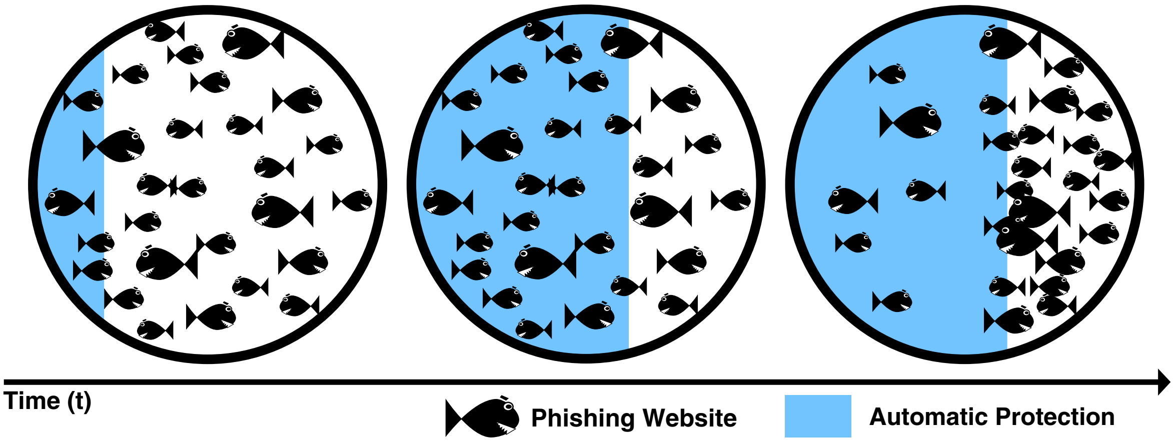


Figure 2.7: With better methods to detect phishing websites the phishers move to areas that are outside of the scope of the automatic detectors. A perfect detection is hardly possible. I argue that HCI research is able to close the gap.

I argue that with the help of HCI research it is possible to close the gap to such an extent that phishing becomes unprofitable. Technical means of detection as they have been used in the past allow the attackers to adapt by changing the technical realization of their phishing website in the future, because this representation is part of the user's perception of the attack. If detection and finally protection uses ways that are built upon the users understanding and perception of an attack the attacker would need to alter the attacks in a way users would be able to notice.

This general idea is independent of today's technology and should persist for new kinds of attacks that could be launched using future technologies of the Internet. For now these changes could make use of new multimedia possibilities introduced in HTML5. As this development cannot be foreseen durable anti-phishing solutions need to be independent of this. Although all projects in this thesis are measured against today's attack styles they have been carefully designed with the idea in mind that for such attacks the user is and ever will be the weakest link.

2.5 Design Space of Current Phishing Attacks

The design space of current phishing attacks is still very broad. While the most common phishing attempt at the moment of writing this might be an impersonation of the online payment provider paypal.com, a huge variety of other phishing attempts for different brands exist as well, together with attacks that even do not target a specific brand (although these are more seldom). In former times phishing emails and websites often already stood out due to bad native language or incorrect brand impersonation attempts. Using the means explained above the quality rose over the last decade.

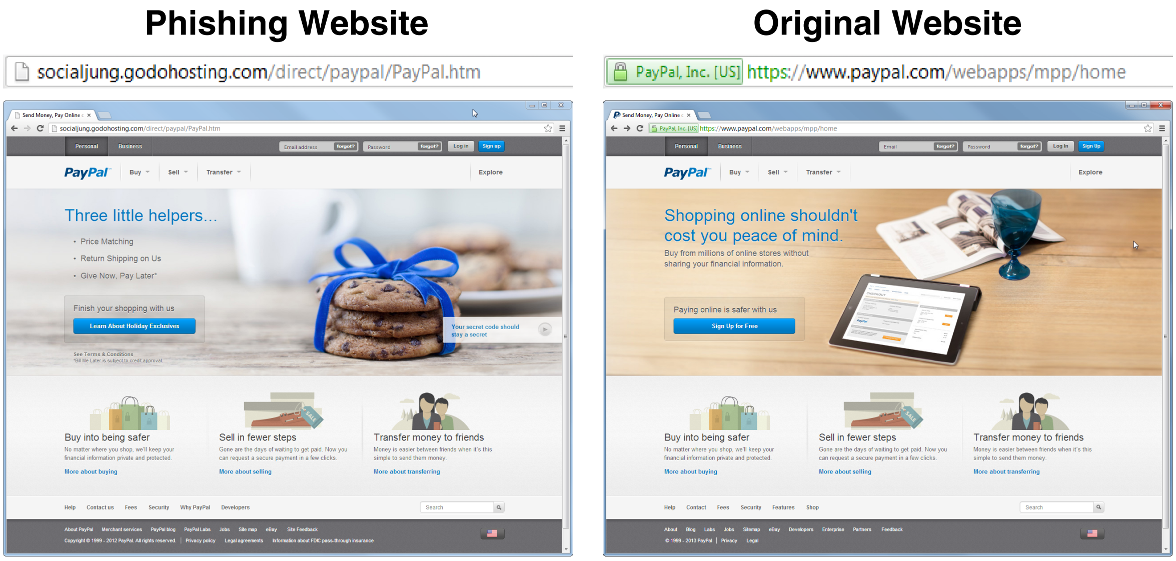

Phishing Website
Original Website
socialjung.godohosting.com/direct/paypal/PayPal.htm
PayPal, Inc. [US] https://www.paypal.com/webapps/mpp/home
Three little helpers...
Price Matching
Return Shipping on Us
Give Now, Pay Later
Shopping online shouldn't cost you peace of mind.
Buy into being safer
Sell in fewer steps
Transfer money to friends

Figure 2.8: Paypal.com phishing website and the original website. Both in a Google Chrome browser accessed at 12th March 2013. The teaser image and text used on the phishing website is a few days older than the teaser image used at that exact same day on paypal.com

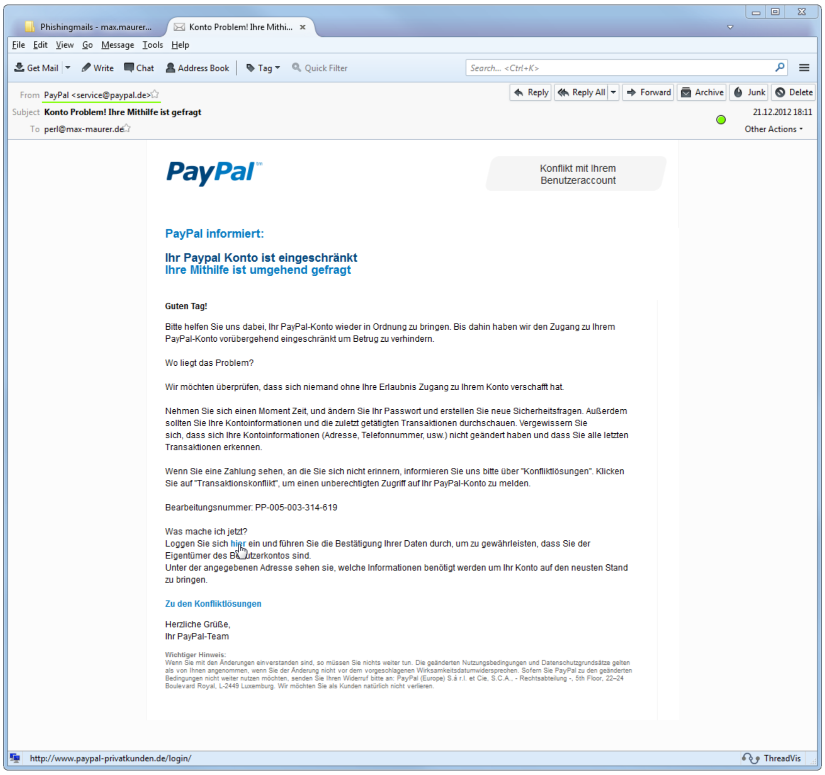
2.5.1 Typical Phishing Examples

To get a proper understanding of how a typical phishing attack today may look like, I want to focus on two different concrete examples of phishing attacks found today.

PayPal Phishing Attack

Figure 2.8 shows the screenshot of a phishing attack of paypal.com and the original website of paypal.com at the same day. The screenshots have both been taken with the most recent version of the Google Chrome Browser on Windows 7. Besides a different teaser image the websites content area is absolutely identical. The different teaser image is most probably due to the fact that this picture was used by the original website at the time when the website was captured and prepared by the phishers. From the website contents alone it would be impossible to recognize the malicious website as a phishing web page but looking at the URL one can clearly see that the phishing website is not served over a secure connection and it has a completely different URL and domain. Still the phishers included the brand name as a folder and as the name of the document that is loaded to make the URL look somehow convincing.

A respective email that points a user towards such an attack may look just like the one in figure 2.1. The email talks about the fact, that the user account is limited and that the users help is needed to resolve the issue. The email states a fake ticket-ID to make it more trustworthy and the sender address is faked to appear as “service@paypal.com”. In fact, for the original paypal.com similar scenarios exist. For example, if an original PayPal account is accessed



From PayPal <service@paypal.de>
Subject **Konto Problem! Ihre Mithilfe ist gefragt**
To perl@max-maurer.de

21.12.2012 18:11

PayPal

Konflikt mit Ihrem
Benutzeraccount

PayPal informiert:

Ihr Paypal Konto ist eingeschränkt
Ihre Mithilfe ist umgehend gefragt

Guten Tag!

Bitte helfen Sie uns dabei, Ihr PayPal-Konto wieder in Ordnung zu bringen. Bis dahin haben wir den Zugang zu Ihrem PayPal-Konto vorübergehend eingeschränkt um Betrug zu verhindern.

Wo liegt das Problem?

Wir möchten überprüfen, dass sich niemand ohne Ihre Erlaubnis Zugang zu Ihrem Konto verschafft hat.

Nehmen Sie sich einen Moment Zeit, und ändern Sie Ihr Passwort und erstellen Sie neue Sicherheitsfragen. Außerdem sollten Sie Ihre Kontoinformationen und die zuletzt getätigten Transaktionen durchschauen. Vergewissern Sie sich, dass sich Ihre Kontoinformationen (Adresse, Telefonnummer, usw.) nicht geändert haben und dass Sie alle letzten Transaktionen erkennen.

Wenn Sie eine Zahlung sehen, an die Sie sich nicht erinnern, informieren Sie uns bitte über "Konfliktlösungen". Klicken Sie auf "Transaktionskonflikt", um einen unberechtigten Zugriff auf Ihr PayPal-Konto zu melden.

Bearbeitungsnummer: PP-005-003-314-619

Was mache ich jetzt?
Loggen Sie sich hier ein und führen Sie die Bestätigung Ihrer Daten durch, um zu gewährleisten, dass Sie der Eigentümer des Benutzerkontos sind.
Unter der angegebenen Adresse sehen sie, welche Informationen benötigt werden um Ihr Konto auf den neusten Stand zu bringen.

Zu den Konfliktlösungen

Herzliche Grüße,
Ihr PayPal-Team

Wichtiger Hinweis:
Wenn Sie mit den Änderungen einverstanden sind, so müssen Sie nichts weiter tun. Die geänderten Nutzungsbedingungen und Datenschutzgrundsätze gelten als von Ihnen angenommen, wenn Sie der Änderung nicht vor dem vorgeschlagenen Wirksamkeitsdatumwidersprechen. Sofern Sie PayPal zu den geänderten Bedingungen nicht weiter nutzen möchten, senden Sie Ihren Widerruf bitte an: PayPal (Europe) S.à r.l. et Cie, S.C.A., - Rechtsabteilung -, 5th Floor, 22–24 Boulevard Royal, L-2449 Luxemburg. Wir möchten Sie als Kunden natürlich nicht verlieren.

http://www.paypal-privatkunden.de/login/

Figure 2.9: German email that leads to a paypal.com phishing attack. The email talks about the fact that account access is limited and that the user needs to help in resolving this conflict.

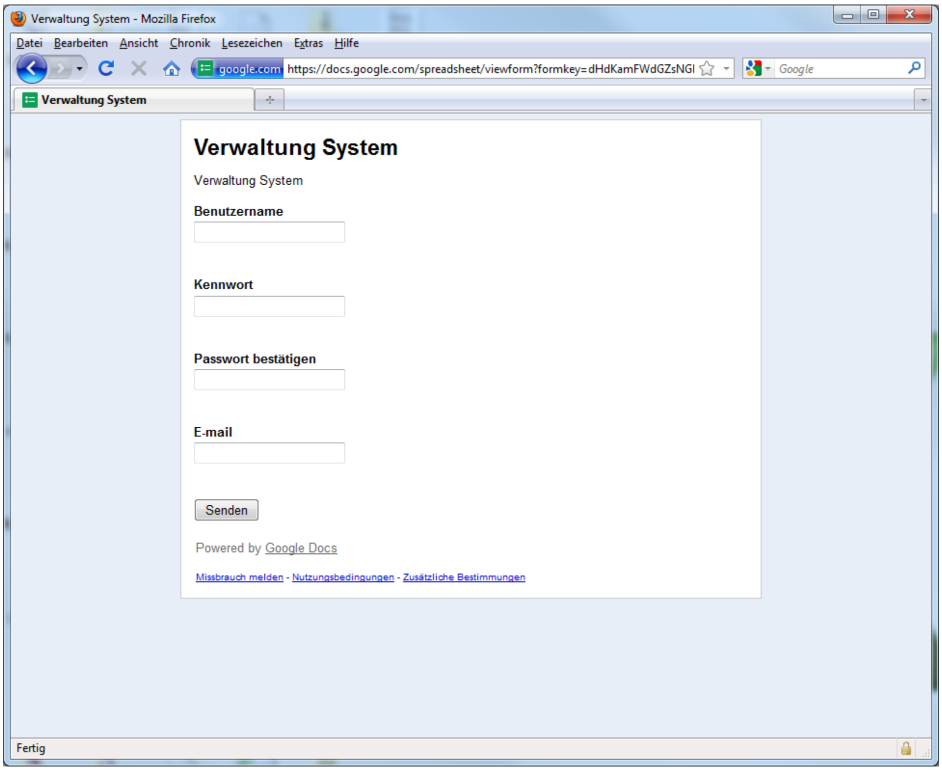
from a country other than the residence country of the user PayPal accounts sometimes really are put into a limited state. From the visual side again, it is nearly impossible to find out that this email is fake except for the fact that it does not contain any personal information about the person addressed. As a security measure PayPal usually adds personal details like the name of the user to such emails. When hovering the link in the email a user can detect the fake URL that in this case points to: “www.paypal-privatkunden.de/login”.

Form Phishing Attack

Compared to the almost perfect phishing attack just explained a few other less professional phishing attempts still exist. These attacks most often just try to get access to an email address getting the email address and the users password. As many users reuse their password for many different services they can then use this data to login almost everywhere. In case this does not work the attackers can still use the password retrieval functions of almost any website to reset the password of an account.

In the current example the pishers even did not setup their own phishing website or hosting. They just used the publicly available methods of Google Docs[9] to create free online surveys.

[9] http://docs.google.com


Verwaltung System - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
google.com
https://docs.google.com/spreadsheet/viewform?formkey=dHdKamFWdGZsNGl
Google
Verwaltung System
Fertig

Verwaltung System

Verwaltung System

Benutzername

Kennwort

Passwort bestätigen

E-mail

Senden

Powered by Google Docs

Missbrauch melden - Nutzungsbedingungen - Zusätzliche Bestimmungen

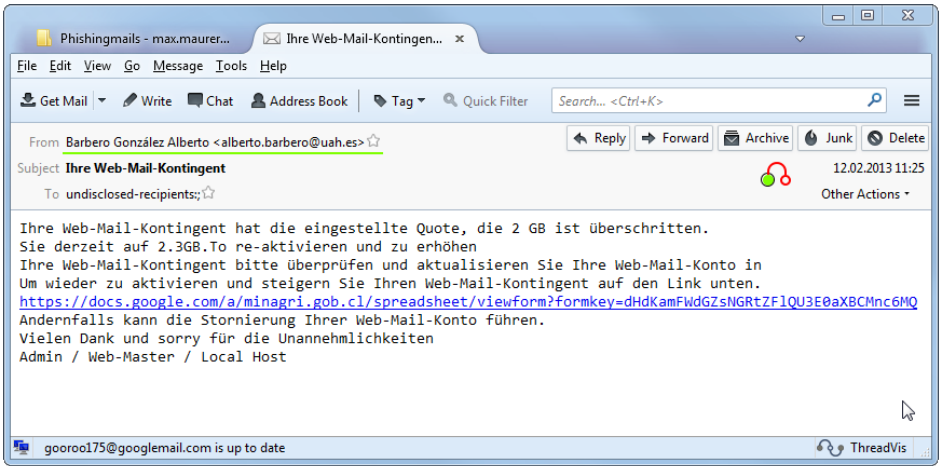
Figure 2.10: Form phishing example original URL from 12th February 2013

Figure 2.10 shows the form loaded inside a browser. The form just asks for username, password (twice) and the email-address of the user. The form headline states "management system" in German with a minor grammatical mistake. The respective email to the user can be seen in figure 2.11. It looks as it has been sent from a private Spanish email address and just uses plain text. In this case the contents talk about the fact that the email inbox of the user exceeds the assigned quota. Again the quality of the German language is somehow poor.

These are just two specific phishing attacks of today, but how exactly does today's phishing landscape look like? For one of the projects carried out for this thesis a large test set of phishing websites was used. I report the quantitative findings and give an overview over several thousand phishing websites in section 5.1 of this thesis.

2.6 Looking at Today's Browsers: Security Indicators in Use

As described in the previous section, it is close to impossible to detect a good phishing attack just by looking at the content area of the browser. This area can be freely designed by



From Barbero González Alberto <alberto.barbero@uah.es>
Subject **Ihre Web-Mail-Kontingent**
To undisclosed-recipients:;
12.02.2013 11:25

```
Ihre Web-Mail-Kontingent hat die eingestellte Quote, die 2 GB ist überschritten.
Sie derzeit auf 2.3GB.To re-aktivieren und zu erhöhen
Ihre Web-Mail-Kontingent bitte überprüfen und aktualisieren Sie Ihre Web-Mail-Konto in
Um wieder zu aktivieren und steigern Sie Ihren Web-Mail-Kontingent auf den Link unten.
https://docs.google.com/a/minagri.gob.cl/spreadsheet/viewform?formkey=dHdKamFWdGZsNGRtZFlQU3E0aXBCMnc6MQ
Andernfalls kann die Stornierung Ihrer Web-Mail-Konto führen.
Vielen Dank und sorry für die Unannehmlichkeiten
Admin / Web-Master / Local Host
```

gooroo175@googlemail.com is up to date

Figure 2.11: The email sent to attract users for the form phishing attack (shown in figure 2.10).

any website that is visited and hence can adopt to any given style. The rest of the browser interface can be modified by the loaded HTML-content only in a limited sense and thus offers a range of indicators related to the security of a user. The only possibility for an attacker to forge these indicators is to display a completely fake browser inside the content area [75]. Ye et al. [330] systematically show how such an attack would work. However, users often are confused about the security indicators as websites misuse them inside their own content areas to generate trust [276].

In this chapter I will give a short summary of the types of indicators that can be found in current websites and how different browsers handle those. This chapter explains the standard indicators (SSL icons, https, SSL ceritifcate infos) and so on. They will be needed later on in projects that try to find better alternatives.

Figure 2.12 shows and overview of how today's browsers look like when they are on a secure SSL encrypted website and how they look like on a phishing website. Most of the security indicators focus on the fact whether the connection to the website is encrypted or not. In most browsers all this information is combined in the location bar of the browser where the user can enter a web address.

Figure 2.13 shows the different indicators in more detail. The encryption based indicators are the https-protocol indicator, the padlock icon and the site identity button (that offers the possibility to open the not always visible site identity dialog). The only security indicator for users that does not rely on encryption is the URL highlighting done in many browsers. Encryption itself does not guarantee that the connecting party is trustworthy. It only tells the user that the connection is protected from eavesdropping (or man-in-the-middle attacks). Yet, nearly all of the phishing websites do not use an encrypted channel (due to the additional effort of registering SSL certificates). Each browser handles and interprets the different indicators a little bit different.

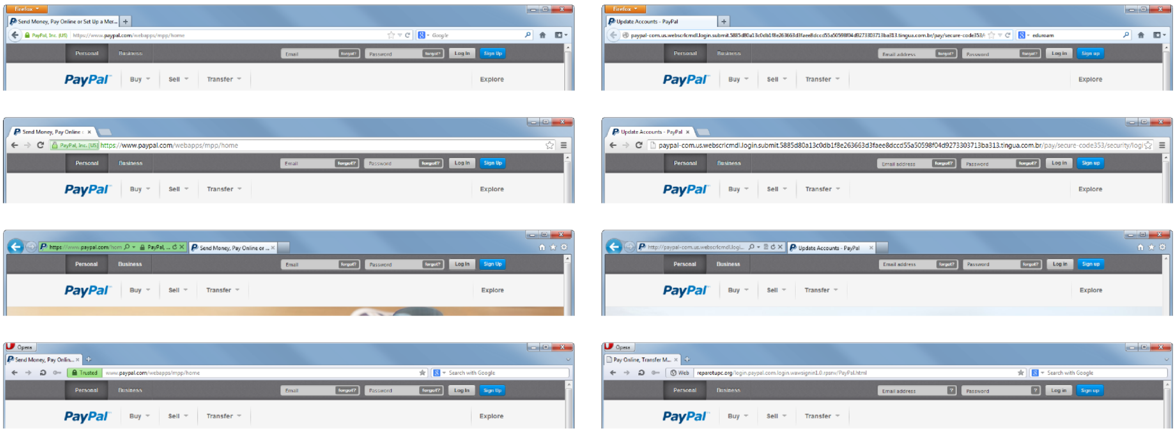
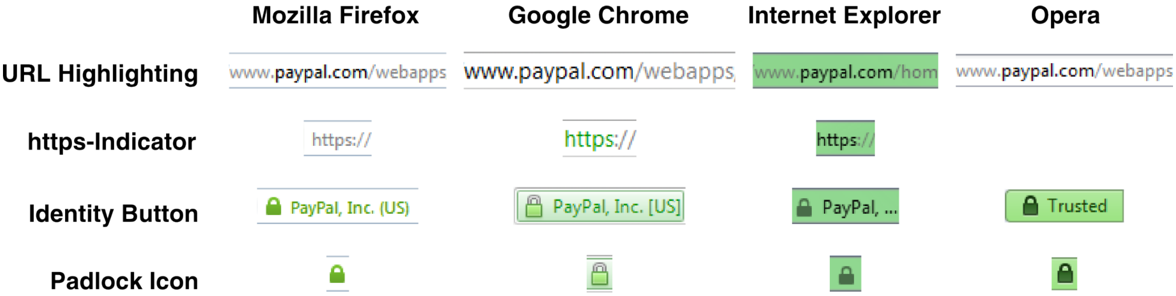
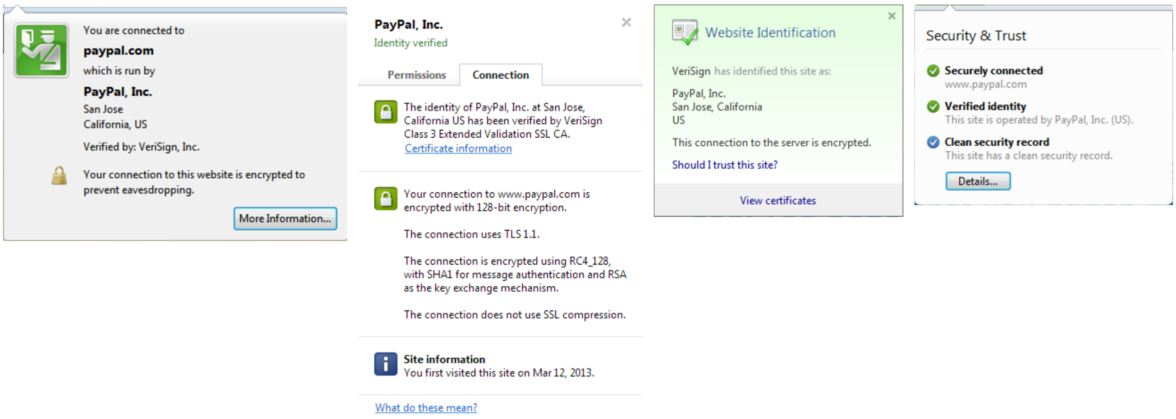


Figure 2.12: The different security indicators of different versions of a browser on a properly encrypted and trusted website (left column) and on phishing websites (right column). From top to bottom different browsers are displayed: Mozilla Firefox, Google Chrome, Microsoft Internet Explorer, Opera. A larger version of this figure can be found in the appendix in figure A.4.



	Mozilla Firefox	Google Chrome	Internet Explorer	Opera
URL Highlighting	www.paypal.com/webapps	www.paypal.com/webapps	www.paypal.com/hom	www.paypal.com/webapps
https-Indicator	https://	https://	https://	
Identity Button	🔒 PayPal, Inc. (US)	🔒 PayPal, Inc. [US]	🔒 PayPal, ...	🔒 Trusted
Padlock Icon	🔒	🔒	🔒	🔒

Figure 2.13: The different security indicators of today’s browsers and how they look like in different browsers.



You are connected to
paypal.com
which is run by
PayPal, Inc.
San Jose
California, US

Verified by: VeriSign, Inc.

Your connection to this website is encrypted to prevent eavesdropping.

More Information...

PayPal, Inc.
Identity verified

Permissions | Connection

The identity of PayPal, Inc. at San Jose, California US has been verified by VeriSign Class 3 Extended Validation SSL CA.
Certificate information

Your connection to www.paypal.com is encrypted with 128-bit encryption.

The connection uses TLS 1.1.

The connection is encrypted using RC4_128, with SHA1 for message authentication and RSA as the key exchange mechanism.

The connection does not use SSL compression.

Site information
You first visited this site on Mar 12, 2013.

What do these mean?

Website Identification

VeriSign has identified this site as:
PayPal, Inc.
San Jose, California
US

This connection to the server is encrypted.
Should I trust this site?

View certificates

Security & Trust

Securely connected
www.paypal.com

Verified identity
This site is operated by PayPal, Inc. (US).

Clean security record
This site has a clean security record.

Details...

Figure 2.14: After pushing the security information button on a web browser different security dialogs appear. From left to right: Mozilla Firefox, Google Chrome, Microsoft Internet Explorer, Opera.

- **Protocol Indicator:** A web browser is able to fetch data using different protocols (e.g. HTTP and FTP). A URL is preceded with such a protocol prefix such that the browser knows how to talk to the foreign server. All web browsers assume HTTP as the standard protocol if nothing is entered by the user. In this case all traffic is transmitted unencrypted and all network nodes that receive the packages of this connection are able to read the package contents. If a connection should be encrypted the HTTPS-protocol is used. This protocol prefix tells the browser to establish an encrypted connection. A user hence can recognize an encrypted connection looking at this prefix. For user convenience many of today's browsers generously hide the protocol prefix of an URL (especially for HTTP). Only the Internet Explorer still displays this prefix for HTTP connections. In case of an encrypted connection Firefox, Chrome and the Internet Explorer display the https-prefix. Opera is the only exception where this protocol prefix is never displayed.

- **Padlock Icon:** The padlock icon is used to indicate an encrypted connection since the early days of web browsers. It used to be part of status bar below the main website window but research showed that this indicator is especially overlooked [59, 67]. The most recent browsers hence brought the icon up and made it a part of the location bar or rather the site identity button.

- **Side Identity Button:** Every encrypted website has a certificate associated with it that holds the keys to set up an encrypted channel and other information about the issuer of this certificate. For the special case of extended validation certificates even more information about the company that acquired the certificate is available. The site identity button appears in the location bar as soon as this information is available and shows some preliminary information about the connected party. In some browsers the button is even present on unencrypted connections and can also be clicked there. Opera is the only browser that does not display the name of the trusted company and instead only displays the text “Trusted” on its site identity button. Once clicked the site identity button brings up the site identity dialog. Usually the site identity button is placed leftmost in the location bar before the URL itself. Only the Microsoft Internet Explorer reserves a small space on the right side of the URL.

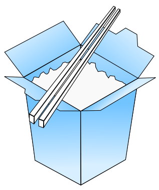
- **Site Identity Dialog:** This dialog is usually hidden from the user and has to be explicitly opened by clicking on the site identity button in the browser. Depending on the browser the opening window displays different information about the encrypted connection or the website that is visited in general. Figure 2.14 shows the site identity dialogs of different browsers. All browsers at least display some information about the trusted party the user is connected to. In some cases, additional information about the encryption itself (at least using an icon [288]) or information about the user's visiting habits to this website are included. In many cases this dialog offers access to even further technical information or explaining resources about the contents of the dialog.

- **Status Color Coding:** In addition to all the aforementioned indicators the browsers also use color coding to display the status of the connection. A connection encrypted using an extended validation certificate is usually displayed in green. For standard SSL certificates some web browsers use different colors (Firefox uses blue for example) whereas other browsers don’t make any difference in color coding for the different certificate types. In some cases only the site identity button changes its color; the Internet Explorer changes the background color of the whole location bar and Google Chrome for example also changes the color of the protocol indicator.

- **URL Highlighting:** The only security indicator of the URL bar that is present for all kinds of connections (no matter of the encryption state) is the URL highlighting. Browsers colorize the different parts of the URL differently to ease the understanding of those. The most important issue here is to highlight the basedomain[10] of the website, because this part really tells the user to which server she is connected. Some phishers try to use subdomains to cover the fact that their website is not the real website the user is visiting creating URLs like “www.paypal.com.fake.org". The real domain name here of course is “fake.org” the other parts of the domain are just subdomains. With domain highlighting this should be easier to recognize. However, Lin et al. [166] showed that this method does not work well, as the URL bar itself is generally overlooked by users.

The current state of the art of security indicators is similar in today’s major browsers with some small differences for the different components that have been mentioned above. Another major difference can be found in figure 2.14: to maximize screen real estate for webpage display, the Microsoft Internet Explorer shares the same horizontal area for website tabs and the location bar. This reduces the size of the location bar and makes it hard to fit all important information into it. Only a short portion of URLs can hence be seen by the user and information on the site identity button is cropped after a few characters. For web browsers on mobile devices the situation is even worse. To use the small size of display best, browser manufacturers hide as many connection details as possible [12]. This makes it nearly impossible for the user to spot changes in URL or encryption during mobile browsing.

Take Home Messages



- ➥ **2.1 What is a Phishing Attack?:** There are numerous definitions of phishing. Some more user-centered other more general. This thesis uses a definition that has the phishing website as its central element.

- ➥ **2.2 The Need to Counteract:** Counteracting phishing attacks is important. Not only because of the loss of money but also because of a potential loss of trust of users into the Internet as a whole.

[10] The base domain is the part of the domain actually identifying the server. Usually this is the top-level-domain (e.g. .org) plus the next preceding part of the domain (e.g. paypal). For “logon.security.paypal.com” the base domain would hence be “paypal.com” (see the project in subchapter 5.4 for more details).

- ➥ **2.3 Phishing Attack Overview:** Although a root cause for phishing lies in existing Internet protocols, developing and deploying new ones is nearly impossible and would potentially come with huge privacy issues.

- ➥ **2.4 A Brief History of Phishing and a Possible Future Outlook:** The term “Phishing” goes back to password “fishing” (probably taken from a computer malware called “FISHES”) combined with the “ph” of the term “phone phreaking”.

- ➥ **2.5 Design Space of Current Phishing Attacks:** From a perfect visual copy to a one-minute survey with spelling mistakes; all kinds of phishing websites still do exist but the majority of the attacks is well made and the number of such attacks is ever increasing.

- ➥ **2.6 Looking at Today’s Browsers: Security Indicators in Use:** Most browsers use six different types of security hints: URL highlighting, https-indicator, site identity button, site identity dialog, padlock icon and color coding. Only the first indicator has nothing to do with the encryption state of the website.

Chapter 3

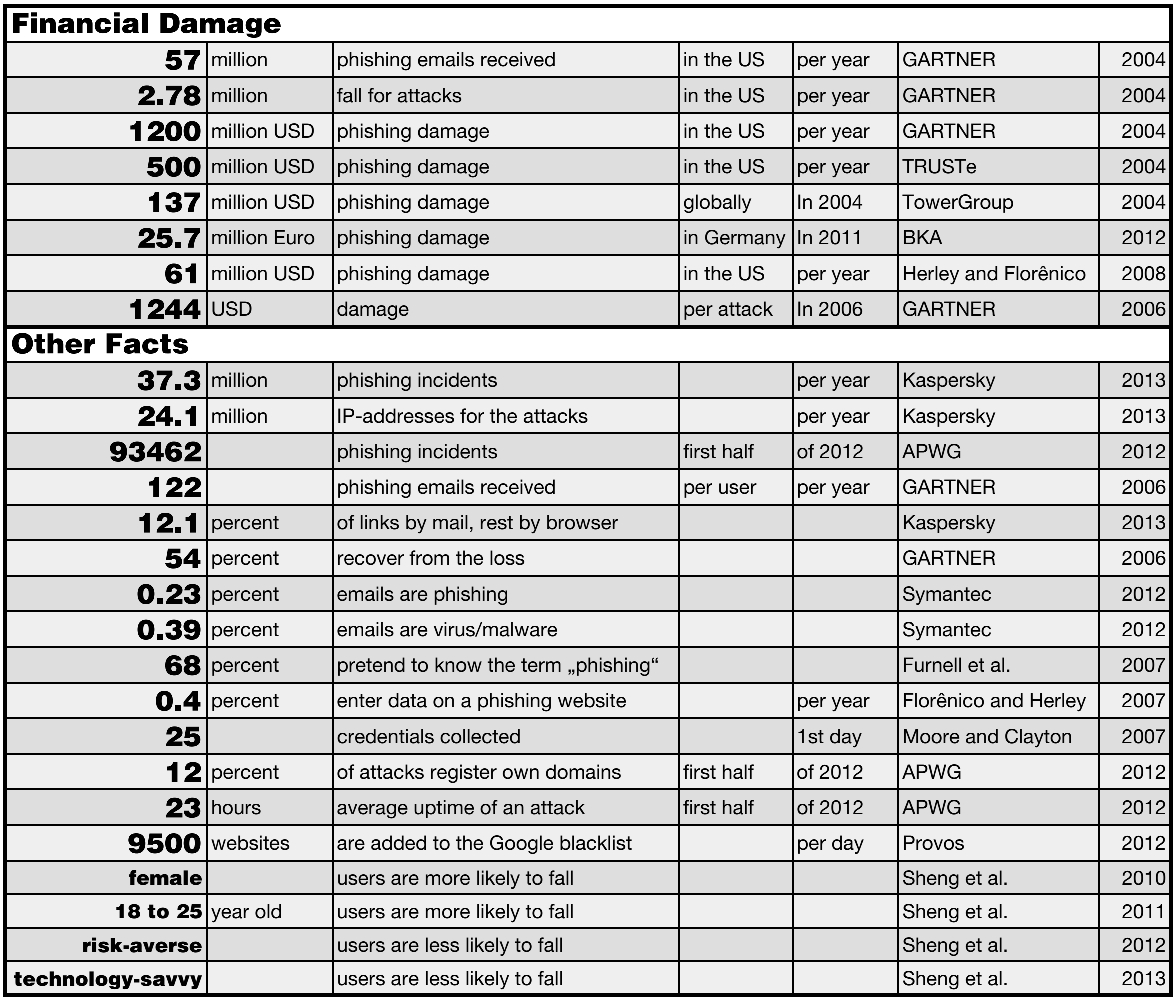
Related Work

Research literature on the topic of phishing and usable security is very diverse. Given the fact that this thesis tries to unite two different research approaches to the topic – detecting and reporting phishing by the use of HCI – related work of both fields has to be considered.

This chapter hence is organized as follows: Section 3.1 first gives an introduction into detailed properties and cost of phishing attacks. Researchers and marketing research companies have collected a lot of details on the problem of phishing and investigated on how and why people fall for it. In section 3.2 the currently deployed concepts and the need for further research in this area are discussed. Section 3.3 looks at the reasons why current user interfaces and warning dialogs fail. How does a computer warning differ from real world warnings and what research has been conducted in both fields? Section 3.4 is dedicated to the controversial topic of phishing education. Some researchers hope that user education is the answer to the problem whereas others claim that user education dedicated to security is pointless. The next two sections (3.5 and 3.6) elaborate on previous research done to enhance either detection or user intervention. Another important aspect for all kinds of work in this area is the way evaluation is carried out through user studies. Section 3.7 reports about this.

3.1 The Phishing Problem

This section contains an in-depth overview on the severity of the problem of phishing. First actual numbers about phishing losses and other parameters of phishing attacks are reported, followed by related work that deals with the question why people are falling for those attacks.



Financial Damage						
57	million	phishing emails received	in the US	per year	GARTNER	2004
2.78	million	fall for attacks	in the US	per year	GARTNER	2004
1200	million USD	phishing damage	in the US	per year	GARTNER	2004
500	million USD	phishing damage	in the US	per year	TRUSTe	2004
137	million USD	phishing damage	globally	In 2004	TowerGroup	2004
25.7	million Euro	phishing damage	in Germany	In 2011	BKA	2012
61	million USD	phishing damage	in the US	per year	Herley and Florênico	2008
1244	USD	damage	per attack	In 2006	GARTNER	2006
Other Facts						
37.3	million	phishing incidents		per year	Kaspersky	2013
24.1	million	IP-addresses for the attacks		per year	Kaspersky	2013
93462		phishing incidents	first half	of 2012	APWG	2012
122		phishing emails received	per user	per year	GARTNER	2006
12.1	percent	of links by mail, rest by browser			Kaspersky	2013
54	percent	recover from the loss			GARTNER	2006
0.23	percent	emails are phishing			Symantec	2012
0.39	percent	emails are virus/malware			Symantec	2012
68	percent	pretend to know the term „phishing“			Furnell et al.	2007
0.4	percent	enter data on a phishing website		per year	Florênico and Herley	2007
25		credentials collected		1st day	Moore and Clayton	2007
12	percent	of attacks register own domains	first half	of 2012	APWG	2012
23	hours	average uptime of an attack	first half	of 2012	APWG	2012
9500	websites	are added to the Google blacklist		per day	Provos	2012
female		users are more likely to fall			Sheng et al.	2010
18 to 25	year old	users are more likely to fall			Sheng et al.	2011
risk-averse		users are less likely to fall			Sheng et al.	2012
technology-savvy		users are less likely to fall			Sheng et al.	2013

Table 3.1: Compiled list of facts about phishing attacks in numbers. Sources: [38, 92, 102, 105, 125, 141, 146, 163, 167, 203, 213, 239, 243, 246, 261]

3.1.1 Phishing in Numbers

To get to know more about the phishing problem in general it is important to look at different aspects of phishing and the numbers that have been generated by researchers and research companies. Perhaps the most important number that comes into mind when thinking about phishing is the average yearly loss that consumers have to endure because of phishing attacks. The estimated numbers here are very diverse ranging from millions to billions. But besides this there are other interesting facts covered in this section. Table 3.1 gives an overview about the most important numbers mentioned in this section.

Monetary Loss Through Phishing

Estimating the global loss due to phishing is very hard as only a small number of incidents can be used to extrapolate the whole global costs. Myers [212] defines three different types of phishing costs. "'Direct costs" that attribute for the total value of money that really is stolen; "indirect costs" incur to users and companies due to phishing attempts (e.g. costs for consumer information) and finally "opportunity costs" where the act of phishing changes the behavior of users such that they for example might not use online banking because they are too afraid. In this section only estimated numbers on "direct costs" will be reported.

In a 2004 publication of Gartner research [167] they estimate that 57 million US adults receive a phishing email per year (30 million being "absolutely sure"). They further estimate that 19% (11 million) click on such a link and 3% (1.78 million) even remember giving personal information to the attackers – one million more unreported incidents are estimated. In fact this results in an estimated loss of 1.2 billion US dollars per year in the US only. A Gartner report of 2005 mentioned in InformationWeek [148] reports a little lower figures with a total US loss of 929 million dollars. In 2006 a followup report [105, 190] reports an increased average of 1,244 USD stolen per attack (257 USD in 2004) and an average number of 112 phishing emails per year and consumer. They also report that only 54% of the consumers recovered from the attacks.

Another 2004 report from the TowerGroup [246] has estimated that the global amount is only 137.1 million USD. John Leyden [162] references this number and compares it to another global value of a survey by TRUSTe amongst 1,335 US Internet users that reports 500 million dollars global loss [163].

In an online interview [117] an 18 year old hacker called "lithium" claims to be phishing since the age of 14 and having stolen over 20 million identities on his own. He claims that social networking sites with people of 14 years and upwards are the best targets for phishing. The number of accounts phished by him each day is reported to be 30,000. Selling the data to scammers he claims to make 3,000 to 4,000 USD. Besides proxies and traffic encryption he uses egold as a payment provider to not get caught.

More recent numbers are mentioned in the key findings of the Javelin Identity Fraud Report 2013 [141]. They claim that 21 billion US dollars have been stolen which is a rising number compared to the years before but lower than the all-time-high of 47 billion US dollars in 2004.

In contrast to the aforementioned sources some researchers think that the phishing problem and the loss of money is largely overestimated. Herley and Florêncio [125] argue that as the money available to phishers is a limited resource the possible revenue for each phisher decreases with the number of attacks increasing. This results in an economic curve that at some points reaches its "equilibrium" where phishing attempts are not lucrative anymore. The authors think that phishing already passed this point and is on the decline instead of the rise. They argue that survey-based measurements overestimate because of a range of introduced biases. In their estimate 0.37% of the web users are phished each year with only half of those really loosing money. Applying an estimated loss of 200 USD per attempt they end up with a total loss of 61 million USD per year.

More Phishing Facts and Figures

The Pingdom[1] website monitoring company each year compiles a report about the "Internet in Numbers". In its 2012 report [235] they mention a total number of 144 billion email messages that are sent per day and the fact that 0.23% of those are phishing in nature. Detailed numbers on this are found in a Symantec intelligence report [213]. 26% of these emails originate in the UK whereas 20% originate in the US. The phishing website locations are diverse: Here 54.7% are hosted in the US with Germany on the second place hosting 4.8% of the phishing websites. 56% of the websites created, use automatic toolkits whereas banking websites are by far the most attacked sector with 38.6%.

A huge source for concrete data on phishing attempts is the APWG (already mentioned earlier). From time to time they publish a survey on phishing statistics they have gathered. In the most recent one available [243] Rasmussen and Aaron present a lot of details on phishing attempts that happened in the first half of 2012. Their report is based on 93,462 incidents that appeared on 64,204 different domains using 202 different TLDs[2]. This means that per domain on average 1.46 phishing attacks are hosted. About two percent of those attacks use IP-addresses instead of domain names whereas 12% make use of maliciously registered domain names. The rest of the attacks are hosted somewhat differently (e.g. at free hosting services). Overall 486 different institutions/brands have been attacked. Concerning the homograph attack (mentioned earlier) only 58 of the websites make use of IDNs[3], hence the number of homograph attack domains must be less than 0.09%. About 80 percent of the attacks are carried out using hacked servers and only some attacks are hosted on registered domains or free web space. The number of phishing URLs that are "disguised" using URL shortener services is small (only 0.005%). The time that phishing websites stay online is getting shorter each year. For the attacks mentioned here the average uptime of an attack was 23 hours and 10 minutes.

When measuring those times, most of the time a certain bias is introduced as researchers do not collect the malicious links from own emails they received but rather use collection sites like PhishTank. Using the appearance of a phishing site in such a list as the time of appearance of a phishing website at all may be wrong. The phishing site could have been online for quite while until it is reported to the respective directory. A correct measurement for the time of appearance of a website would usually be the point in time when the phishing website is first made public (e.g. by sending the first phishing email containing the link). Determining this point in time is nearly impossible.

[1] www.pingdom.com

[2] A top-level-domain (TLD) is the most important part of a domain name (e.g. .com, .org, .de) [237].

[3] IDN stands for internationalized domain name and denotes domain names that make use of glyphs or other non-standard characters from the unicode character set. In fact a IDN is registered using a specific notation that uses only ASCII characters. This notation is called "Punycode"[4] [83].

[4] In Punycode notation special characters are basically skipped and replaced by a dash at the end of the original string. Finally the missing characters are prepended encoded to be readable by a state machine [56].

In Germany the Bundeskriminalamt also keeps track of Cybercrime incidents in an annual report [38]. For 2011 they report 6,422 phishing incidents and losses of 25,7 million Euro.

Niels Provos from the security team of Google Safe Browsing [111] reports that 9,500 malicious websites are added to their index each day [239]. About one third of these entries are phishing attacks [112] the rest are malware websites.

A very recent report from the Kaspersky Labs [146] sees phishing still being on a major rise (87% up in one year). In their 2013 report they saw phishing attacks on 37.3 million users within one year. The number of distinct attacking IP-addresses also rose to 24.1 million. That means that there is one IP-address available for less than two attacked users. According to their report Germany is after the US and the United Kingdom the third most attacked country in the world. The most interesting finding from this report is the fact the the attack channel of email links only makes up 12.1% of the attacks whereas the majority of 87.9% is spread out using other websites. CTO Nikita Shvetsov reports that phishing is now a separate and clearly visible threat [256].

Moore and Clayton [203] report in their work about web server statistics they gathered from real phishing websites. They drew many conclusions from this data. The most important one perhaps being that approx. 25 victims enter sound data on a phishing website the day it is first set up and 10 more data sets follow with each day the same page remains online.

3.1.2 Who is Falling for Phishing and Why?

As mentioned in the beginning security is never the primary goal [296] but is this fact enough to account for all the people falling for phishing attacks? In fact many of the phishing baits sent out even catch users by saying that they are about the security of the user's account. This subsection elaborates on different research works that have looked at the issue of why people fall for phishing and why phishing attacks seem sound to them.

"Why Phishing Works" by Dhamija et al. [67] is one of the first and most cited papers for research about user behavior and phishing. In a study they showed 20 phishing and non-phishing websites to 22 participants and let them determine which ones were fraudulent. In one case more than 90% of the participants were fooled by an attack. From this data they deduced three major problems why users fail to detect websites: A "lack of knowledge" about computer systems and security indicators is on the one hand a problem why users are not able to judge attacks correctly. Using "visual deception" the phishers craft texts, images and look and feels that are so close to the original ones that the users are unable to notice any difference. In combination with "bounded attention" of the users they fail to notice the presence or absence of security indicators. More recently in 2011 Erkillä [82] summarized related studies of the past years and adds another important property to the list. The strong feeling of the user to be secure in the browser and to be in control of what happens make users ignore security warnings more easily. According to Papachristos [229] the visual properties seem to be extremely strong as user judgments about website parameters like “visual appeal”, “usability” or even “credibility” are made within fractions of a second.

Downs et al. [72] asked 232 participants of a survey about their behavior towards certain suspicious emails, websites and URLs. People that were able to correctly define the term role play, where significantly less likely to fall for the attacks. They also found out that the level of perceived severity of possible consequences cannot be used to predict participant behavior.

Sajano and Wilson [273] looked more generally at the reasons why people fall for scams and report eight principles as reasons for that: the distraction – users taking care only for what they want to do, neglecting the security components around that procedure – and the social compliance principle – people should not question authority – perhaps being the ones most important for phishing.

Rick Wash [291] looked at the folk models that people make up about security and hackers. According to his research people see hackers as “digital graffiti artists”, “burglars who break into computers” or “contractors that support criminals”. He argues that the models that users make up about attackers should be taken into account when creating security help and advice. He also gives a list of possible pieces of advice and matches those to his folk models.

Similar to that Friedman et al. [98, 99] looked at how people understand things like “secure connections” and what they fear about security on the Internet. They compared rural to suburban to high-tech users and found interesting differences. E.g. technology-savvy users are more afraid of information loss (92%) than rural users (54%).

Downs et al. [73] conducted interviews with 20 non-expert computer users to understand their decision-making when encountering suspicious emails. The participants saw eight emails in a roleplay within-subject study and were asked how they would behave with each of the emails (three emails were non-phishing). The authors report three strategies that make people fall for phishing: “This email appears to be for me”, “It’s normal to hear from companies you do business with”, “Reputable companies will send emails”. Asking for the cues that make the users suspicious 95% mentioned a spoofed “from” address, secure site lock icons (85%), broken images on a web page (80%) and unexpected or strange URLs (55%).

Florêncio and Herley [92] have collected a lot of data about password usage on different websites by using data gathered through the Microsoft Windows Live Toolbar in a period of approximately three months. Results show that in average 6.5 different passwords are used for 25 accounts that a user has in average. Stronger passwords are reused more seldom than weak passwords and per day a user types in average 8.1 passwords. Concerning phishing, the authors used a by-URL analysis that was cross-referenced with a list of known phishing attacks to show that annually at least 0.4% of the users submit a password to a phishing website.

Furnell et al. [102] looked at the security perceptions of 415 personal Internet users in 2007. Although 93% of the participants of their online survey rated themselves to be intermediate or advanced level users only 20% are “very confident” with the security of their home computer. Concerning different types of threats, “phishing” is by far the least known threat. Only 68% claimed to understand this security related term – compared to at least 83% understanding for all other terms.

In a study with 398 subjects rating emails and websites according to their degree of phishiness Tsow and Jakobsson [284] discovered that URLs can change authenticity ratings and that the overuse of security advice can have negative effects.

Mannan and van Oorshot [176] looked at the problems that users face especially with online banking and its guidelines. Whenever a bank tells their customers to watch out for suspicious login URLs but uses a long and complicated login URL themselves it gets hard for the users to follow the advice. Similar problems occur with guidelines about SSL indicators (like the lock icon). Asking users about their behavior not even all of them (93%) sign out at the end of a banking session.

To better understand the problems that phishing threats pose to the user Dong et al. [70] tried to model user-phishing interaction. They also argue that throughout the user’s decision making process to a take an action or not the perception construction is most important as this defines whether a user will fall for an attack or not. A misperception occurs due to three different reasons: insufficient information, misinterpretation and “incomplete expectation perception ability drop”.

In 2010, Sheng et al. [261] looked at phishing susceptibility of users more closely. They showed phishing emails to 1001 mechanical turk[5] users and did a regression analysis on different demographic properties afterwards. The results showed that female users are significantly more prone to fall for phishing as male users. The same is true for the age group of users between 18 and 25. Participants that rated themselves as more technologically knowledgeable and more risk averse people were less likely to fall for phishing attacks.

Blythe et al. [33] did an online survey with 224 respondents to measure their ability to differentiate between spam and phishing emails. The users succeeded in detecting an average of 7.2 of 10 phishing emails whilst misclassifying 5.7 of 10 genuine emails as phishing. The authors also found that including a logo in the phishing email makes a significant difference.

3.2 The Current State of Detection Methods

Current web browsers and other computer software like anti-virus software are already trying to protect the computer users from the phishing problem using various ways described in this section. Some of these methods have their roots in economic product development rather than research. Whatever measures taken so far they are still not yet protective enough and the phishers find ever new ways of avoiding the detection methods. This chapter will first

[5] Mechanical Turk is a platform dedicated to offering companies a platform to have humans work on human intelligence tasks they get paid for. Such tasks can also cover the participation in research experiments [10, 299].

look at the most common protection method of black- and whitelists moving on to additional security toolbars that can be installed in the browser. After that the possibilities of virus scanners, typo checkers and law enforcement are discussed before taking a look at whether a complete change to the Internet architecture could possibly solve the phishing problem.

3.2.1 Black- and Whitelists

Blacklists are the main source of phishing protection that today's browsers offer to their users [260]. The browsers match a visited website URL against a list of known malicious websites and disable the access to that website using a warning screen. The Mozilla Firefox and the Google Chrome browser both use the "Google Safe Browsing environment" [111] while Microsoft's Internet Explorer has a so called "SmartScreen Filter" [194]. In case of the Google Safe Browsing environment [111] for example two different types of protection exist for privacy reasons: An "enhanced" mode looks up every entry in an online blacklist whereas the second mode maintains a local blacklist of phishing pages for privacy reasons. To enforce this detection a single "phishing warden" exists for the application monitoring all HTTP requests to determine whether the request is blacklisted. URLs are not sent unencrypted over the network: instead a shared secret is generated using a HTTPS connection which is then used to encrypt the URL traffic to the blacklist server.

In fact blacklist based solutions today can be completely sidelined by attackers. This can be accomplished by completely personalizing each URL sent to the users. If the URLs are then blocked on basis of the URLs each submitted URL will only belong to one attacked user and it will be useless to add it onto a blacklist once it has been detected. Using IP-address based blocking can also be subverted by phishers using "fast-flux networks" [203]. In such cases page requests are handled not by a single computer at a specific IP, instead an army of computers is used, swapping the DNS records of a website to ever new computers and IP locations. Even when storing the captured credentials phishers have invented a clever tactic by storing the information gathered on other online services encrypted or hidden in a way they cannot be found (e.g. embedded in images) [140]. This makes it possible to retrieve the information even if the machine hosting the attack is taken down.

Sheng et al. [264] did an empirical analysis of phishing blacklists using 191 fresh phishing reports being less than 30 minutes old. During their tests most of the blacklists caught less than 20% of the attacks initially. Even after 48 hours the coverage of all blacklists was still below 90%. However, blacklists do work well concerning false positives. For a second test set of 13,458 legitimate URLs not a single URL was accidentally reported as being phishing.

A recent study by NSS Labs [219] looking at how current browsers are capable of detecting 754 samples of "malicious software" showed a major improvement of the Internet Explorer browser to detect 83.17% through URL reputation but but finding another 16.79% through application reputation for the given test set. The report does not cover extensive data about false positives.

At least for a large number of today’s attacks blacklists still seem to be the right answer but could cease to work, once more attackers have adapted to the blacklist protection. Jung and Sit [144] have examined DNS[6] blacklists for spam filtering and found that the traffic produced by spam and blacklist lookups is increasing and that the number of blacklists entries between blacklists can differ to up to 87%.

3.2.2 Security Toolbars

The list of existing toolbar protection approaches is long and Stepp and Collberg [277] tried to summarize them. They cluster toolbars into different categories. “Information-oriented tools” report more information to the user than would usually be present (e.g. “SpoofGuard”, “SpoofStick”, “Trust Toolbar”). “Database-oriented tools” rely on database information being maintained on other servers and are dependent of this backend that could possibly take better decisions (e.g. “Cloudmark”, “eBay Toolbar”). Finally they talk about “Domain-oriented tools” relying on the fact that malicious websites have nearly no possiblity to fake the original URL of a website (e.g. “PwdHash”, “RoboForm”).

In 2007 Zhang et al. [335] evaluated a large list of those toolbars and other existing anti-phishing tools including the blacklists used by Firefox and Google as well as some toolbars dedicated to user security (e.g. the eBay toolbar). For a total of twelve different tools they evaluated which percentage of phishing websites are correctly identified and how those detection rates evolve as time passes by. Depending on the method used, between 28% and 91% of the phishing websites were initially discovered by the tools. Tools having a high detection rate usually had a lot of false positives (92% for the tool that detected 91% of the phishing websites). This shows that these tools do not work perfectly. In an interviews series conducted with 31 security experts Sheng et al. [262] compiled 18 recommendations for phishing countermeasures of the future. The fifth recommendation addresses the detection rate of such tools:

> *Web browser vendors should continue to improve the performance of integrated browser antiphishing warning systems, with a goal to catch 90% of phishing URLs within an hour after they go online.”*
>
> **– Sheng et al. 2009 [262] –**

3.2.3 Virus Scanners

Virus scanning software more and more tries to protect their users from any security problem anywhere on their computing hardware. To accomplish this they even monitor the network

[6] DNS is the Domain Name System that is used to built up and resolve Internet Domain Names to IP-addresses [202].

traffic going to the computer's web browsers and try to disable problematic traffic. The measurements taken are similar to the ones used by toolbars. Blacklists are used as a first line of defense usually combined with some HTML heuristics afterwards [253].

3.2.4 Typo Checkers

For web browsers additional plugins exist that protect the users from mistyping URLs [226, 227]. This can also protect the user from similarly spelled phishing domains. The problem here is that it is hard to determine whether the different domain name has not been typed on purpose. On the other hand, some of the plugins only work for hand-typed URLs and not for ones that have been clicked in emails or for links on websites. In those cases the phishing protection is lowered to a minimum. The research question in how far URL typos can be used for phishing detection is tackled by one of the projects of this thesis in section 5.4.

3.2.5 Law Enforcement and Website Takedown

As with classical real world criminals the creators of phishing attacks should be tracked down by law enforcement agencies. Due to the global nature of the Internet and the speed at which phishing attacks are brought up and taken offline, this is close to impossible.

More importantly website and hosting operators need to take down a phishing attack once it occurred on a system. Although the average online time of a phishing already is at a historic low of 23 hours (as reported above) [243] Moore and Clayton [204] examined the impact of website takedown. They found that the first day of an attack usually yields the largest number of credentials by examining website logs of real phishing websites. However, the numbers of credentials entered does not drop to zero after a longer period of time. In one example a phishing website that was online for more than a month still received a steady stream of phishing responses. This shows that although website takedown cannot be a final answer to the problem it needs to be carried out as fast as possible. Sheng et al. [262] also have proposed some recommendations for law enforcement. They suggest to improve international cooperation, provide law enforcement with better capabilities and to get corporations to submit more fraud data to law enforcement agencies.

3.2.6 Changing The Internet Architecture

As already mentioned earlier the core of the problem lies in the Internet architecture that allows the avoidance of methods mentioned in this chapter and allows to spoof websites of other companies (see section 2.2). However, changing the Internet architecture is close to impossible. Hartman [119] has proposed changes to the architecture that would make phishing impossible. The overarching goal would be to protect against the fact that confidential information is disclosed to any parties that are not allowed to receive the data. According to Hartman more sophisticated login methods like smartcards should be supported; it needs to be possible to have a trusted user interface for websites. Another necessity Hartman proposes would be that a website does never receive a "strong password equivalent" (as it is today). He proposes to have passwords that are used to authenticate but are never transmitted in full to the authentication partner. If the partner would not be the intended recipient he would then be unable to reuse the received token for login at the original website. Most of Hartman's propositions (in case they could be implemented and deployed) target authentication processes. The implementation of those would hence not protect the user from phishing of other confidential information. Similar to this RSA[7] [252] proposes the use of two-factor authentication to counter phishing attacks. In fact a proper use of one-time passwords[8] for example can help to avoid the successful logon of attackers. Again this would only protect authentication credentials and implies and additional security hurdle users have to pass. Oppliger and Gejek [225] discuss other possible protection methods on the browser and on the client side.

3.3 The Current State of User Intervention

To this point the lack of phishing protection has been presented to be mostly due to a lack of proper detection methods. But once a suspicious website is found the way of proper intervention[9] comes into play. This other side of the coin is equally important. This chapter first looks into general warning research from the "offline world" that has a long and detailed tradition and then moves on to warning research in the computer and more specifically the web browser space to find out why warnings do not work as intended by their creators.

3.3.1 Classical Warning Research

Creating warnings in the "offline world" is very different than it is on the screen. On the one hand the number of possible parameters that can be changed for a warning is much larger (e.g. material, size, colorspace, actively lit); computer warnings are limited to the same pixel and color space that all other display elements occupy. On the other hand computer warnings can be much more flexible including dynamic elements or even changing contents.

[7] RSA is an american network security company selling authentication tokens [311].

[8] One Time Passwords are additional security tokens that can only be used once. For another authentication attempt another password is needed. This kind of authentication has the advantage, that a password that has been captured by an attacker is useless to him as it can not be used for a second authentication attempt performed by the attacker. To work as intended the range of possible one time passwords should be large and the duration of validity for each password should be short [308].

[9] As defined in the introduction of this thesis the term "intervention" is used here for any means a system (mainly a browser) takes to inform the user of a risk or to avoid the entering of critical information by the user.

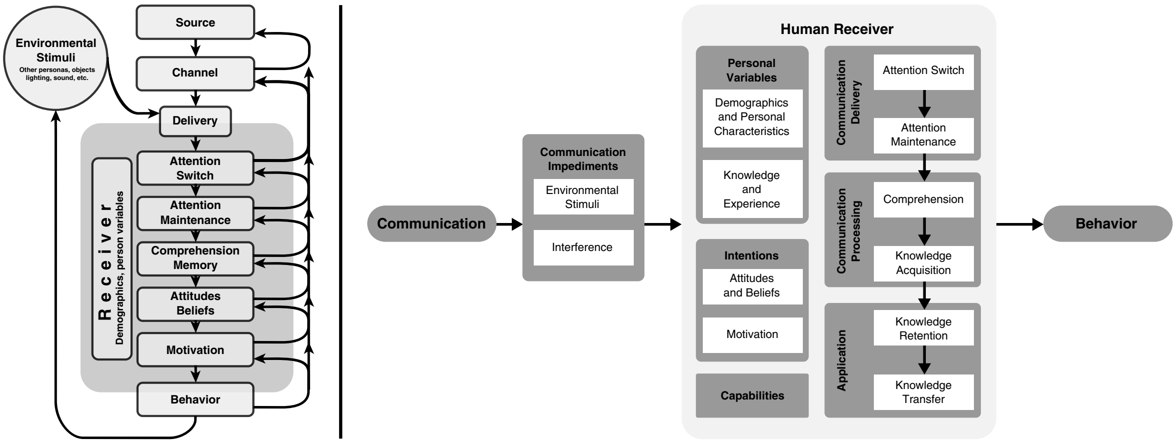


Figure 3.1: Left: Communication-Human Information Processing Model (C-HIP) [315] [redrawn]; **Right:** the enhanced Human in the Loop Model (HITL) [60] [redrawn]

Michael Wogalter has compiled a major collection of warning research in his "Handbook of Warnings" [316] that gives a good introduction into general warning literature. In the scope of this book he defines a warning as follows:

> **Warning** *Warnings are safety communications used to inform people about hazards so that undesirable consequences are avoided or minimized.*
>
> **– Michael Wogalter 2006 [317] –**

Besides, warnings have four important functions: they are used to communicate safety information, should be able to influence peoples' behavior, prevent injury and property damage and finally serve as a reminder [317].

As a surrounding basis for warning research Wogalter presents the Communication-Human Information Processing Model (C-HIP) [315] (see left side of figure 3.1). It starts with a source of warning information that is translated in several stages to finally end up in some specific form of behavior. Inside the receiving person a variety of parameters influence the final resulting behavior: attention switch (can the warning attract attention?); attention maintenance (is it really examined?); comprehension (is the meaning understood?); beliefs and attitudes (with which the warning has to comply) and finally the motivation to really carry out the intended behavior. The whole model is designed as a loopback process between the different stages that also takes environmental stimuli into account. Although not primarily made for computer warnings it matches perfectly to the problems that occur with the perceptions of computer warnings.

Lorrie Faith Cranor [60] extended this model (see right side of figure 3.1) to fit even more closely to computer security with the goal that user security problems can be explored during the design phase of new software. She also also applies her model to anti-phishing tools to show how her metrics could be applied there.

In classical warning research the methodology is similar to what will be used later on in this thesis. It makes use of self-reported data through interviews and questionnaires but also measures quantitative data using head movements, eye tracking and response time [268] (see section 3.7 for more details).

Together with active warnings (bound to sensors) the problem of false alarms came up [31] (e.g. smoke detectors). This false alarm effect is also called the *cry-wolf phenomenon*. It understands both the human receiving the warning and the sensor reporting it as detectors and defines responsiveness as a function of both elements. A lot of different theories about this way of signal mistrust have been set up by warning researchers and several mediators have been found that have an impact on warning mistrust: signal urgency, hearsay information, signal stimulus modality, signal reaction mode and information redundancy. To get rid of the problem the detection threshold can be adjusted, operators can be trained to handle false alarms or the workload of checking the warning could be optimized.

Another major problem with security warnings – similar to the cry-wolf phenomenon – are habituation effects. Users get used to those warnings and always perform a standardized action with each warning to get rid of it. Amer and Maris [11] conducted an experiment with 88 participants to prove this. The participants had to enter sales data into a computer form. After submitting each data block a warning dialog was presented that needed the choice of “yes” or “no” to enter the data. The measured reading times of the warning declined quickly from 15 seconds in average to about 2 seconds. After the eighth exposure of the warning a different warning dialog was presented that looked similar to the first one but required an inverted answer. In this case only 11% of the users spotted the change and pressed the correct option of the warning dialog.

Wogalter et al. [320] see computer displays and the world wide web as a chance for better warnings rather than a problem, offering more flexibility for displaying data. Within this thesis we can see that the possibilities gained through computer screens are not used well and the problem of false alarms is even bigger with computers than it is in standard warning scenarios.

3.3.2 Computer-Specific Warning Literature

Research how users perceive computer security and its warnings has also started early and forms the basis of the research on usable security. One of the most fundamental papers was written in 1999 by Alma Whitten and J.D. Tygar [296]. Whilst testing users for their understanding and ability to use PGP 5.0[10] the majority of the users failed. The authors derived different "problematic properties" from the experiment, the most important one being "the unmotivated user property" in other terms that security is a secondary goal to users.

In 2004, Dourish et al. [71] found users being frustrated towards security. They perceive security as a barrier and want it to be either delegated towards technology or other individuals or organizations. Users don't use technical means to apply security but rely on other methods like obscuring (e.g. by not mentioning exact details in an email conversation where the communication partner can understand the details from the context). Windows user account control (UAC)[11] for example is not correctly applied by 69% of the users [210].

Cormac Herley [123] thinks that this rejection of security by the user is rather rational because the overall costs of security cannot cover the benefits that arise by conforming to security. He uses password rules, certificate errors and security teaching to prove his point.

Bravo-Lillo et al. [34] built a detailed mental model of user behavior to warning messages. They differentiate between novice and advanced users stating that novice users have a binary understanding of warnings (either that their computer is infected or the warning is not actually a problem). Advanced users are able to judge the safety of their actions before engaging in them and consider more factors for their decisions.

This shows that computer warnings in general already pose a problem to the users' understanding and will to obey them. Other literature has looked specifically at warnings inside the browser meant to protect the users from phishing. Whalen and Inkpen [294] already looked at eye-tracking data in browser security in 2005. In two phases they observed the viewing behavior of participants during web browsing with and without security priming. Without security priming not a single instance of security checks on any security indicator was found at all, even though the participants had to carry out tasks involving seemingly confidential information. This shows that indicators like the https-protocol indicator or the lock-icon simply are not noticed. After security instructions 25% of the participants still did not look at any security indicator. The most examined indicator was the lock icon (69%).

Bardzell et al. [21] describe important issues of the human context of phishing. They emphasize that compliance with security protocols is not automatically done by users; it depends on their perception of the risk that may happen. A lack of understanding of security and privacy problems poses a problem that users cannot understand the security indicators they nevertheless do not notice.

Tyler Close from the W3C [53] identified some general problems for web browser warnings today. A poorly defined area for the browser chrome – the area of the browser surrounding the website contents – makes it possible to fake web-browser indicators, for example by using picture-in-picture-attacks (displaying the picture of a non existent browser within

[10] PGP stands for "Pretty Good Privacy" and is a computer software made for cryptographic data handling primarily used in email communication [310].

[11] UAC is used by Windows operating systems from Vista upwards to limit standard application rights. If needed the user can elevate a software to administrative rights by confirming a security dialog [196].

the browsers content area). Despite, some of the information in the chrome can already be altered by an attacker to the degree that the user cannot easily spot the difference (e.g. subdomains instead of real domain names). As a last point Close mentions the bad user understanding of the chrome area. Besides the fact that picture-in-picture attacks are not very often used the browser chrome technically is the most trustable area of the browser user interface. I will make use of it for example in the project in subchapter 5.6.

In section 3.2.2 we already saw that the detection rate of security toolbars is not optimal, but even more important is the fact whether users would be able to take notice of suggestions made by the toolbars. Wu et al. [324] presented 30 subjects with 20 different emails, five of which were phishing attacks. They created three different toolbars to see how many people would fall for attacks despite being warned by the toolbars' indicators. In average 52% of the attacks were successfully showing that security toolbar warnings don't work well yet.

Egelman et al. [78] evaluated web browsing phishing warnings two years later. They compared two actively blocking warnings (Firefox, IE), to passive warnings (not blocking the interaction with the website) and a control condition with no warning at all. 70 participants were assigned to four different warning conditions and were presented with spoofed emails that arrived at their personal inbox after purchasing two items on shopping websites. All but two people fell for the emails and clicked on the phishing link but the first active warning was able to prevent all of the users from entering data on the phishing website. For the second active warning still only 45% of the users entered their data. Looking at the passive warning and the control condition 90% of the participants entered their data, showing that passive warnings are more or less useless in this context.

Sunshine et al. [280] looked at the effectiveness of SSL warnings. In their study, 409 survey respondents were asked about how they would behave for certain SSL warning situations. They showed that for warnings that are harder to ignore people that don't understand the warnings are more likely to heed these. People that understood the warnings acted depending on the type of warning. In this experiment no attacking websites were used. The warnings were shown on original websites and hence could have denoted man-in-the-middle attacks.

3.4 Phishing Education

Security education is a highly controversial topic among researchers and although many companies and agencies offer security advice and learning material on phishing [37,222,287] the general idea of phishing education contradicts the fact that security is never the users primary goal [296] to some extent.

Jakob Nielsen [217] and Stefan Görling [113] see user education as being the failure of developers to come up with systems that are secure enough by design. He argues that with the help of HCI, security must not be "added" but "should be integrated with the users anticipated behaviour". Besides, the level of education and the actual behavior of the user are not necessarily the same thing. However, after interviewing 31 security experts, Sheng et al. [262] found that most of them agree that education and awareness are phishing countermeasures that need to be emphasized more.

In contrast to Nielsen and Görling, Sheng et al. [263] showed that phishing education using the playful approach of a game does a better job in teaching people about phishing URLs than tutorials or other training materials do. Playing a game called *anti-phishing phil*, users arrived at 87% correctness rate when classifying URLs between phishing and non-phishing. Existing training material (74%) and a crafted training material (80%) performed worse. Nevertheless all materials increased success above the baseline of about 65%. With concepts like this several companies offer the service of training employees towards phishing. Besides Anti-Phishing-Phil being offered as a product [321] other companies like phishme.com [232] even offer the possibility to deliberately spear phish employees for educational purposes. They claim to have successfully "phished and educated" over 3.1 million people.

Also in 2007 Kumaraguru et al. [155] designed a phishing email training system that is supposed to be used within the standard email reading tasks of a user. Within this study the researchers also found that a comic strip as teaching material is better suited than textual notices or a text and graphics intervention. In another work Kumaraguru et al. [156, 157] looked at how retention and transfer of phishing education can be enhanced. Participants received three sets of emails. One before training, a second one immediately afterwards and a third one, one week later to measure the success of training and its retention. Using embedded training material the immediate and delayed correctness rates where much higher. Sheng et al. [261] combined Anti-Phishing-Phil, the comic strip used by Kumaraguru et al. and other training materials in a larger questionnaire study with 1001 respondents. The percentage of attacks people fell for dropped from 47% to 28%. All education materials performed similarly well in this study.

3.5 Research Concepts for Detection

As security is most often taken as a solely technical topic, first security research on phishing also tried to find technical means to counteract successful phishing attempts. This chapter covers research approaches that try to detect or prevent phishing from a technical perspective. Although this thesis focuses mainly on the detection of phishing attempts, a reasonable amount of related work exists that tries to make phishing impossible by designing new types of authentication for example. Some of these concepts are briefly explained in section 3.5.1. When trying to detect phishing websites automatically, the different features that can be used for such a detection are limited: technical connection properties, textual content, visual appearance or combinations of those. Hence the successive sections report about phishing detection methodologies that have been developed on basis of this classification.

3.5.1 General Phishing Defense

A big problem about phishing is the fact that users can enter any kind of credentials at any website. Impostors can collect and store these credentials and reuse them anywhere else. If it would be impossible for the users to resubmit the credentials of another site at a phishing website the problem would be solved at least concerning authentication credentials. Ross et al. [249] created "PwdHash" that does not use the user's actual secret credentials to login to a website but instead creates a hash out of the user's password combined with the domain of the website or other information. Like this the users original password is never submitted to the server. Since the domain of a phishing website differs, the hash value generated with the same password would be a different one and the original user credentials would remain safe. A major downside of this principle is that it already has to be present when registering new accounts. Besides this, Chiasson et al. [48] did a usability study on the system and found out that it significantly reduces usability of the password input sequence. Only 42% of the users were able to successfully mitigate their password towards the new system. A similar system has been created by Yee and Sitaker [332]: "Passpet" allows the user to create and store user credentials for websites by associating a pet name to each website. Upon revisiting the website the credentials for the website are automatically resolved. Passwords here are also generated by a hashing process using a master password for the whole browser. Individual site passwords are transparent to the user.

Other methods try to detect or fool phishing websites by submitting random data to the websites. The concept of Chandrasekaran et al. [42] envisions to retrieve URLs from suspicious emails, visit them in an automatic manner and feeding such a website with artificial credentials. If a website does not properly reject those credentials it could be a phishing attempt that collects this data. "BogusBiter" by Yue and Wang [333] submits a large number of user credentials to suspected websites whenever the user submits her own credentials. The additionally submitted credentials are very similar to the original credentials submitted by the user and the real login credentials are injected somewhere between all other fake credentials. Using this technique it should become cumbersome for an attacker to manually filter the submitted credentials for the real one.

Another set of approaches introduces an additional authentication token that is not controlled by the user itself and can hence not be accidentally entered on a different website. Parno et al. [230] suggest to store a secure credential on the users mobile phone that is then used for every authentication attempt. In case of failure of this additional device the user then would not be able to authenticate anymore. BeamAuth [6] uses a browser bookmark and the fragment part of the URL[12] to store an additional authentication token there. This token can be read by a JavaScript function and can be transferred together with the standard credentials to the server. An attacker would be able to only steal the standard credentials but would miss

[12] The fragment part of the URL is appended at the very end after a hash symbol and is usually used to navigate the browsers viewport to a specific place on the loaded page [27] (e.g. `http://www.website.com/book.html#secondchapter`)

the token taken from the bookmarked URL. The problem here is that a user not having his login-bookmark ready will not be able to authenticate to the website.

Other approaches tend to find phishing attacks by looking at the password entry behavior of a large user base. Florêncio and Herley [91,94] let a central server monitor which credentials are used by which users on which domain (for privacy reasons only hashes of the credentials and domains are stored). If it occurs that different users reuse their credentials from domain *A* on a different domain *B* that is unknown this might denote a phishing attack and although the credentials of those users would have been lost one would still be able to immediately tell which users have fallen for the attack and alert them.

Jakobsson and Myers [138] want to introduce a feedback loop into the password entry process. For each character of a password a user enters on a website, a response image will be displayed by the user, slowly forming a series of those images that should be easily recognizable by the user. If the image does not look like the one the user is used to, she either entered the wrong password or entered her password on a different website (hence a potential phishing website). This could already be detected with the first few characters of a password such that a user could stop revealing the complete password immediately. For this concept the way and the speed of the average password entry are a problem. Entering a password character by character waiting for a server response in between can be very cumbersome, especially as some users don't even bother to look on the screen during password entry.

One final set of concepts tries to preprocess webcontent to make it more trustworthy. Miyamoto et al. [198] want to sanitize malicious web content by cleaning it on a proxy server before sending it to the users browser. Known phishing inputs should simply be removed from the websites before arriving at the user. The ability to correctly differentiate between phishing and non-phishing websites is a requirement needed by their system without stating any possible detection method. Shin et al. [266] propose the reverse by creating "prooflets" denoting signed HTML code that is guaranteed to come from a specific source.

All concepts presented in this section are only useful for the protection of user account credentials. Other credentials or information like credit card data, social security information or bank account data could still be requested by phishers on their websites without being noticed. A general protection concept for all critical information still has to be found.

3.5.2 Detection Attempts for Different Features

Although black- and whitelists are known not to be the final answer to the phishing problem they still are an effective concept in productive use. Hence a couple of researchers thought of ways to enhance the concept. Prakash et al. [238] tried to use predictive blacklisting to find new phishing websites by combining known websites. They discovered around 18,000 new phishing URLs from a set of 6,000 initial blacklist entries. Using five heuristics they generate new URLs (e.g. by replacing the TLD) and validate their existence and their similarity with a known phishing website afterwards. A very similar approach was taken by Felegyhazi et al. [84] who derived 3.5 to 15 new blacklist entries from an existing entry. Instead of a blacklist extension Cao et al. [41] propose to manage an "automated individual white-list" that alerts the user whenever she is about to conduct suspicious logins. The system needs a training phase to build up the initial whitelist for the user and afterwards uses a classifier to distinguish real from fraudulent logon websites. They show that the number of unknown new login websites that appear decreases to nearly zero within a week. Although the general black- and whitelist concepts can be enhanced they still lack the problems of missing or incorrect entries and the problem of a missing zero hour detection. For projects within this thesis black- and whitelists are hence not primarily used. In cases where it is important to reduce the number of false positives a detector has, a whitelist might come in handy (e.g. in one of my projects in section 5.5).

URLs cannot only be used to put them on black or whitelists. A lot of related work takes them into account to compute the probability of a phishing attack. Ma et al. [172, 173] use lexical and host-based features of a URL to train a machine learning classifier. The lexical features are taken from different related work [153, 189] whereas the host-based features contain WHOIS[13] properties – like the date of registration – or domain name properties – like the Time-To-Live[14] of the DNS record –. Using a test set of 15,000 benign and 5,500 malicious URLs they compared different sets of the acquired features and using the full set of features they were able to reduce their error rate to 1.24%. Garera et al. [104] also trained a logistic regression classifier with features to detect four different phishing URL types (IP-address attacks, attacks using the brand name in the path, large hostname attacks, misspelled domains). As features they make use of URL properties stored by Google like the PageRank[15] or a quality score of a website but also use domain based features and try to identify possible obfuscation types. Using their approach they arrive at a false positive rate of 1.2%. Blum et al. [32] use the "confidence weighted algorithm" and base their research on the research by Ma et al. [172, 173] excluding the host-based features but extending the lexical feature set to 25 lexical features. They achieve an error rate of 3%. Within this thesis I also studied the idea of making use of the URL as means to detect phishing. The project in section 5.4 explains how we tried to use the spell checking algorithms of major search engines to detect phishing attempts using the URL.

Moving away from URL and domain name features other properties of websites have also been used for detection. For the forthcoming related work a classification and logical ordering is not easy. Researchers make use of HTML content, machine learning, visual properties of the HTML content, or visual similarity between rendered images of a website in all different kinds of combinations. The following paragraphs start with the more content oriented related work moving to the visual similarity related work. To get a better understanding of

[13] WHOIS is a protocol that allows to query properties of registered domains (e.g. the owner) [314].

[14] Time-To-Live of a DNS record limits the validity of a propagated DNS record before it needs to be reacquired at the host server owning the DNS record [312].

[15] PageRank is an index of the importance of a website measured by Google. The basic algorithm is based on the number incoming and outgoing links a website has. An equal starting score assigned to every website is split up and accumulated for the incoming links [309].

	Authors	Year	Black- or Whitelists	URL / URL Features	Domain-Features	HTML Content	HTML Styles	Visual Similarity	Machine Learning
1	Prakash et al.	2010	✓	✓					
2	Cao et al.	2008	✓	✓					
3	Felegyhazi et al.	2010	✓	✓					
4	Garera et al.	2007		✓	✓				✓
5	Ma et al.	2009		✓	✓				✓
6	Blum et al.	2010		✓					✓
7	Chou et al.	2004		✓	✓	✓			
8	Zhang et al.	2008		✓	✓	✓			
9	Wardman and Warner	2008		✓		✓			
10	Xing et al.	2011	✓			✓			
11	Rosiello et al.	2008				✓			
12	Wenyin et al.	2006							
13	Whittaker et al.	2010		✓	✓	✓			✓
14	Xiang et al.	2011							
15	Dunlop et al.	2010						✓	
16	Liu et al.	2006				✓	✓		
17	Medvet et al.	2008				✓	✓	✓	
18	Huang et al.	2010		✓		✓		✓	
19	Fu et al.	2006						✓	
20	Chen et al.	2009						✓	
21	Chen et al.	2010						✓	
22	Bannur et al.	2011		✓		✓	✓	✓	✓
23	Afroz et al.	2009/2011				✓		✓	

Table 3.2: Different types of detection approaches and their usage throughout the related work of this subchapter. In general the chaper is moving from content based detection methods to visual similarity based detection methods taking publication time into account where possible.

the different methods table 3.2 shows an overview of the different related work mentioned here.

In 2004, Chou et al. [49] presented one of the first combined concepts using URL check for misleading URLs, image checks for crosslinked images, link checks (similar to the URL checks) and password checks (Is a password requested? Does the form use https?) to compute a total spoof score (TSS) of the visited website. Evaluation of SpoofGuard was done rather informally with fourteen malicious websites and some manual false positive testing by the authors.

Zhang et al. [336] use the very simple concept of TF-IDF (term frequency-inverse document frequency) for their detection approach. With TF-IDF the most important words of a document are extracted by comparing the number of occurrences of all words in the document to the occurrences of the words in the language in general. Using a test set of 100 phishing websites and 100 legitimate websites the TF-IDF approach achieved a 94% true positive rate but also had a 30% false positive rate. Hence, the authors added some basic domain and URL features and optimized their detection rate to 97% true positive and 6% false positive.

Wardman et al. [290] try to use MD5-hashing[16] to detect similar phishing attacks. Since small changes in the HTML code of a website would immediately result in a different hash value they use MD5 values of all files a website consists of and compare how many files between two different websites match. They managed to automatically classify 34.1% of a phishing test set they used.

Xiang et al. [329] try to do "soft matching" of website content against blacklist entries using the "shingling"-technique. Using a sliding window approach the content of a certain number of recent Phishing attacks is analyzed and then compared to potential new phishes using n-grams[17] of the website content. They achieve a true positive rate of 70% with their test data together with a false positive rate of 0.12%. This work extended their prior work [328] using TF-IDF to find important terms in websites and process the search results of those terms to validate the currently visited page.

Rosiello et al. [248] extend their AntiPhish-approach [152] with a DOM-Tree[18] component. Within the tag structure of an original and a potential phishing website they try to measure the similarity by finding the longest similar subtree. Choosing a threshold that detects all given Phishing websites in their test set, they end up with a false positive rate of 16.9%.

Other researchers rely on machine learning to find phishing websites. Whittaker et al. [295] from Google Inc. combined features from the page's URL, the hosting information and the page's HTML content. The logistic regression classifier is trained once per day with a sample of 10 million URLs. Using their model they achieve a high true positive rate of 92% and a nearly non existing false negative rate of 0.01%. Besides these findings they show that any website having a Google PageRank of more than 0.5 can usually be neglected as a possible phishing attempt.

CANTINA+ is another feature-based approach from Xiang et al. [327] in 2011. Before using machine learning techniques they prepend a hash-based duplicate remover to filter out websites that are 100% identical and a login form detector to only process websites

[16] Hashing is used to map input information to a different (usually much smaller) output. The hash value then can be used to differentiate between the different inputs. For each hash function it should be very hard to create a different input the produces the same output (hash collision). Such functions can be used as cryptographic hash functions. Inputs close to each other should result in very diverse outputs [301]. The MD5 algorithm is such a function used to generate hash values.

[17] N-grams are small subsequences of a larger text symbol sequence that can be used for certain types of algorithms. Trigrams of words (3-character sub-sequences) are for example used to do language detection of texts [307].

[18] The Document Object Model (DOM) is a convention how nodes in XML, HTML and similar documents can be represented and addressed. The nodes form a tree-like structure that can be traversed to reach each node within the document [159].

containing login forms. Afterwards they use a set of different URL-based, HTML-based and web-based features including the aforementioned PageRank or the age of the domain. They evaluated against phishing websites from PhishTank and legitimate websites from Alexa and the Yahoo[19] website directory. Finally they achieve a true positive rate of 93% together with a false positive rate of 0.4%.

But which machine learning algorithm is best suited for phishing detection? Miyamoto et al. [199] compared nine different machine learning techniques including the often used Support Vector Machines (SVM) and logistic regression. By using the feature set of the classic CANTINA approach (containing eight binary features) they showed that AdaBoost is the machine learning algorithm that performs best. Within this thesis and its project I did not make use of machine learning techniques as it was not required for the respective prototypes. For large classification and optimization task it may still come in handy although in the case of phishing. However, machine learning also follows known algorithms that can be exploited by attackers [22].

Another tier of research started to take the visual style of a website into account by trying to mitigate the effect that Phishers can alter HTML code quite easily without changing the visual outcome of the rendered website. GoldPhish by Dunlop et al. [76] hence applies optical character recognition (OCR) to the screenshots taken from websites visited by the user. Afterwards they submit the captured text line-by-line to a search engine and check whether the indicated website appears in the top four results. With a limited test set their system achieved a true positive rate of 98% and had no false positives.

Moving more towards the visual properties, Wenyin et al. [168, 169] included "style similarity" into their approach (together with layout and block level similarity) taking features like font-family or the background-color of a document into account. The approach was tested with eight phishing websites and 328 original websites. Using a threshold that detects all phishing websites resulted in a false positive rate of 1%. A threshold value generating no false positives failed to detect one of the phishing websites (13%).

Medvet et al. [191] compute a page signature out of text elements and image elements, taking style of the text elements into account but also data of the images (e.g. width and height) as well as the color histograms. Having two different signatures a similarity score can be computed assigning different weights to each individual feature comparison. Using 41 positive pairs (that should match) and 161 negative pairs (that should not match) they achieve and overall false negative rate of 7.4% and a false positive rate of 0%.

Huang et al. [130] included the images on a webpage into their analysis together with URL keywords extracted from the domain name and content keywords found through TF-IDF. From the images on a website they tried to find the logo image by intersecting the images from the different subpages of website. The extracted images are then matched against each other and the result of this matching is combined with the detection results from the text-

[19] www.yahoo.com

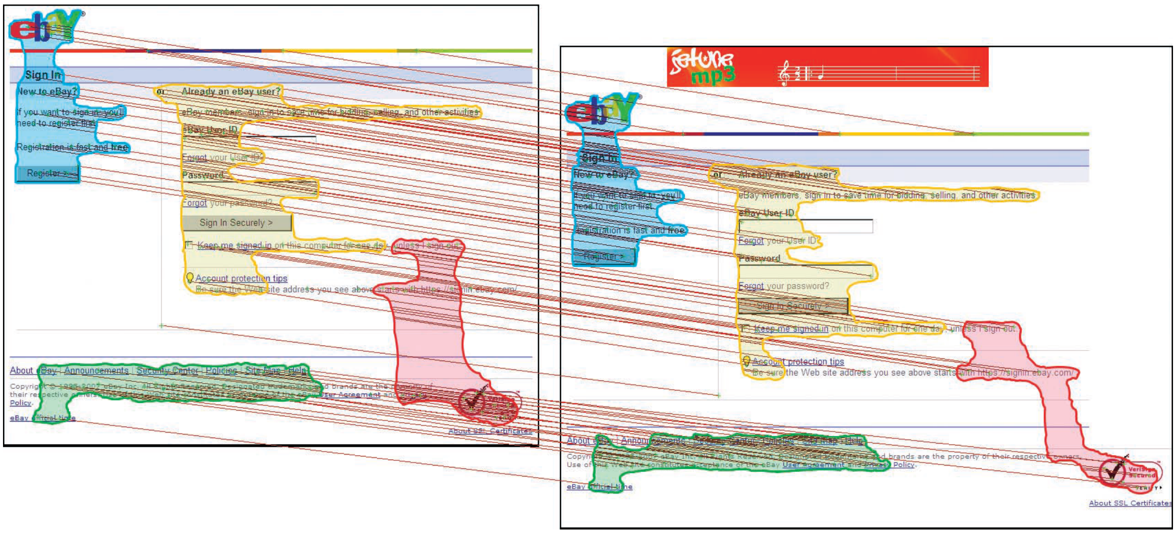

Sign In
New to eBay?
If you want to sign in, you'll need to register first.
Registration is fast and free.
Register >
or
Already an eBay user?
eBay members, sign in to save time for bidding, selling, and other activities.
eBay User ID
Forgot your User ID?
Password
Forgot your password?
Sign In Securely >
Keep me signed in on this computer for one day, unless I sign out.
Account protection tips
Be sure the Web site address you see above starts with https://signin.ebay.com/
About eBay | Announcements | Security Center | Policies | Site Map | Help
eBay official time
About SSL Certificates
mp3

Figure 3.2: Chen et al. [45] use their "Contrast-Context-Histogram" to detect similar websites finding four image clusters and comparing their similarity.

based features. With a test set of 200 phishing and 270 non-phishing websites they achieved a false positive rate of 1% and a false negative rate of 6%.

A first attempt to only use visual similarity between websites was done by Fu et al. [101] using the Earth Mover's Distance (EMD) that is usually used to solve the problem of distributing goods of producers to consumers. After normalizing website screenshots to a smaller size and using a degraded color space of 4,096 different values the images are compared using EMD. The test set used consisted of nine phishing websites compared to 10,272 legitimate URLs retrieved from Google using certain keywords. Training their thresholds with 1,000 websites the remaining websites achieved a true positive rate of 89% and a true negative rate of close to 100%.

Chen et al. [45] use their own "Contrast-Context-Histogram"-feature for phishing detection. It uses neighboring pixels of relative brightness around certain keypoints (see figure 3.2 for an example). Afterwards four clusters are built on the two webpages and compared against each other. They evaluated their concept with "several" phishing websites compared against 300 websites of well-known banks and auction services. Within their test set they achieve a high accuracy between 95% and 98%.

Another team of researchers (also Chen et al.) [47] motivate their comparison approach using gestalt theory[20] thus showing the importance of HCI in the field of phishing research. As a

[20] Gestalt theory reasons that the arrangement of individual visual items as a whole determines the meaning of an image [293].

similarity metric they use the normalized compression distance (NCD)[21]. Using bzip2[22] as a compressor they tested the similarity value gained by 24 pairs of phishing and non-phishing websites showing that the NCD values within a pair are lower – hence more similar – than between comparison with the other websites. In a test with 320 phishing websites for 16 legitimate websites they achieved roughly 95% true positives and 1.7% false positives (here using LZMA[23] as a compressor).

A mix of nearly all possible features is used by Bannur et al. [20]. They use URL properties, structural properties, page link information, semantic information but finally also visual features – in this case spatial properties like the roughness, histogram information of the images, and finally SIFT to identify local visual objects. Using machine learning the features were evaluated against a test set of 60,000 phishing and 120,000 non-phishing websites (70% used for training, 30% used for testing). Including the URL and content-based features they achieved accuracies up to 98%.

Afroz and Greenstadt [7, 8] call their concept to detect phishing websites "PhishZoo". In a first profile making phase profiles of websites that should be protected are extracted. Logo images have to be manually selected. When searching for phishing websites, the existing profiles are compared against potential phishes. The selected Logo image is compared against all images of the website using SIFT. Using 1000 phishing websites and 200 non-phishing websites results in an accuracy of 96% although the image matching takes up to 17 seconds for a single phishing website against all created profiles.

Besides the aforementioned concepts even more related work does exist (e.g. [2, 228, 334]) but it all comes down to new variations of the aforementioned variables. At a first glance this mass of related work in the detection domain might look as if the problem of phishing detection is already solved. However, when looking at today's number of ongoing attacks one can clearly see that the problem is still out there. Besides, there are also some other problems with prior research (not necessarily applying every concept). In some cases of related work there was no real representative test set. Testing a detector with only a couple of phishing websites is not enough no matter how well the approach might seem to work. In other cases a problem of the mixture between HTML-detection and visual detection might arise. Detection properties that are derived from HTML or the URL can often be changed by the attacker without the user noticing. Thus combined approaches that rely on such metrics might loose in effectiveness once the phishers would adapt to them. Other approaches have very fine grained detection approaches but rely on direct comparison between two websites that is very time costly. In a real-world scenario a potential phishing website would need to be compared against millions of websites (phishing or legitimate). If a single comparison

[21] NCD is broadly spoken a measurement for how difficult it is to turn one object into another object. This is usually measured by looking at how well a combination of both objects can be compressed. Similar objects can be compressed to a higher extend [164].

[22] bzip2 is a compressor using the Burrows-Wheeler compression algorithm. Besides recoding of characters to save space a "move-to-front transform" is applied that can afterwards be compressed by other means [300].

[23] LZMA stands for Lempel-Ziv-Markow chain algorithm and is a lossless compression algorithm with a higher compression rate than bzip2 [305].

would take around 17 seconds the user would most probably have died before receiving a result from the detector. In many of the papers the detection time needed is not reported at all. A last problem exists that can be applied to all related work in the detection area, in case there is no 100% certainty about a website being phishing or not, the user would be needed as a last resort to decide whether to visit a website or not. Taking warning science and HCI into account as a second stage of testing is hence a must for every detection algorithm. In one of my projects (see chapter 5.8) I also address the issue of detecting phishing website through visual similarity but in contrast to the aforementioned approaches I try to solely rely on this factor (as it is the only factor based on human perception) and evaluate it together with a warning design as a holistic HCI concept.

Besides the numerous ideas for detection that do not incorporate user behavior at all, one single piece of related work stands out of this crowd. Ronda et al. [247] developed an approach where they try to match the visited website against search term results of the most important words but instead of aggregating those automatically they let the user type in what he thinks the website is about. If a page is visited that has a PageRank of more than five or in case it is a special whitelisted page the warning technique is not triggered at all.

3.5.3 Making Use of a Community

Another option to overcome failures of automatic detection is to add a crowd component to the detection process. Some researchers and tools try to do so though community-based approaches. Experts- and semi-experts may help other users in detecting malicious websites.

A partial-established industry product that supports this kind of protection is "Web of Trust" [322]. Based on their own experiences users can rate any website they visit using four categories: trustworthiness, vendor liability, privacy and child safety. Everyone that uses the WOT browser plugin – 73 million downloads so far — can then access the average ratings of other users. If the average ratings are beneath a certain threshold a popover warning appears whenever a WOT-user visits such a site. Concerning phishing the approach has a basic problem. Generally spoken WOT is not more than a weighted black- and whitelist for known domains. Hence, for unknown phishing attacks no data exists that could be used to rate the website. In addition, it is unknown how far those website ratings of other people can influence the behavior of other users. I addressed this problem in one of my projects covered in section 5.3.

From a research perspective such concepts can be framed using the term "social navigation". This means using aggregated data of decisions or opinions of others to provide users with guidance and enhance usability, security and privacy for them [108]. Goecks et al. [108] looked at problems that may arise using "social navigation" using the management of browser cookies and firewall management as two examples. They propose to use a two step decision process to solve the problem of users always going with the recommended decision without using common sense. DiGioia and Dourish [69] looked at file sharing permissions as an example.

More phishing oriented related work exists, too but has been previously mentioned. The PhishTank (introduced in section 5.1.2) service itself makes use of human helpers to verify the blacklist entries for their phishing blacklist and was evaluated by Moore and Clayton [206]. Participation of users in the PhishTank system is power-law distributed. This is contradictory to the principle that in community-based systems the mass participation could be used to correct biased answers of single users.

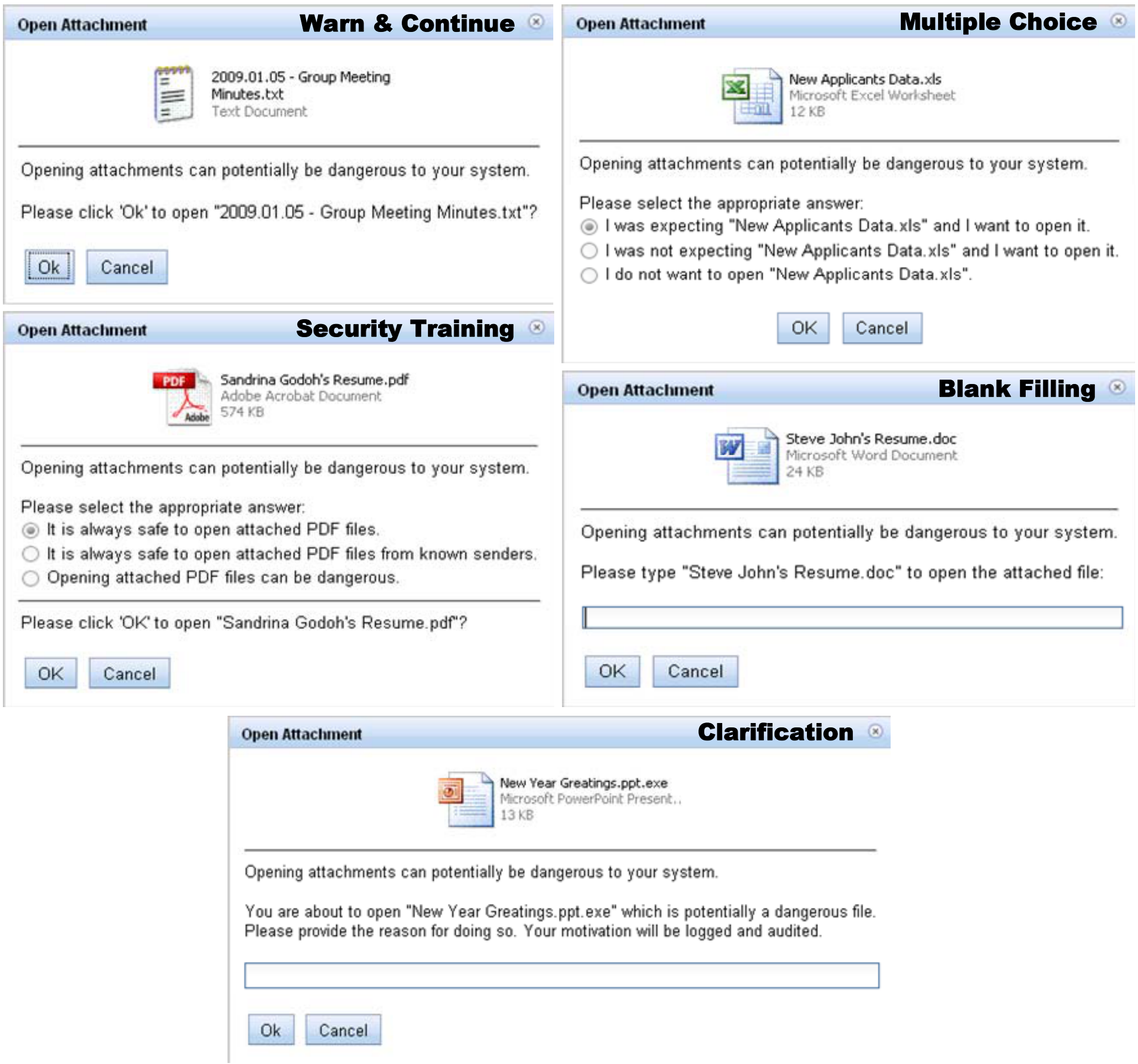
3.6 Research Concepts for User Intervention

Not all security researchers focus on providing new methods for detection or technical prevention. Some also address the user interface problems that come along with any detected problem: How to display the right warning to alert the user of the (potential) problem that is occurring? Security dialogs are known for their habituation effects as proved by Amer and Maris [11] (see section 3.3.1 for details on the different problems). To overcome those problems some solutions have been proposed. These are mostly of a general nature and phishing as an application area is nearly never used. In the first subsection of this chapter ideas of altering dialog contents and options to reduce habituation are explained followed by some related work dealing with how dialog contents should look like in general to achieve the best effect.

3.6.1 Adaptive Dialogs

A new concept called polymorphic dialogs was presented by Brustoloni and Villamarín-Salomón [36] in 2007. They make use of email-attachments to demonstrate their concept. Traditional email software either does not warn the user about risky attachments at all (e.g. Thunderbird) or simply hides the attachment without notifying the user (e.g. Microsoft Outlook). Instead, the proposed polymorphic dialogs change and rearrange their possible answers whenever they appear. In addition they propose to possibly audit the users dialog behavior and for example temporarily suspend the user from working. They evaluated their concept with a role-playing exercise with 20 participants. The additional audition warnings led to users accepting significantly less unjustified risks.

Keukelaere et al. [151] presented the concept of adaptive security dialogs. The basic idea here is to have more complex security dialogs for less experienced users and less complex ones for security experts. This way inexperienced users get a more detailed explanation on their options while experts “waste” less time dealing with the warning. For these adaptive dialogs additional data about the security performance of the user or other environmental data is taken into account. Therefore the authors introduce different new types of dialog boxes (see figure 3.3). Evaluating their concept with 24 participants, people read the new warnings for a significantly longer time and more people did not open dangerous file attachments of simulated email messages.



Opening attachments can potentially be dangerous to your system.

Please click 'Ok' to open "2009.01.05 - Group Meeting Minutes.txt"?

Opening attachments can potentially be dangerous to your system.

Please select the appropriate answer:
● I was expecting "New Applicants Data.xls" and I want to open it.
○ I was not expecting "New Applicants Data.xls" and I want to open it.
○ I do not want to open "New Applicants Data.xls".

Opening attachments can potentially be dangerous to your system.

Please select the appropriate answer:
● It is always safe to open attached PDF files.
○ It is always safe to open attached PDF files from known senders.
○ Opening attached PDF files can be dangerous.

Please click 'OK' to open "Sandrina Godoh's Resume.pdf"?

Opening attachments can potentially be dangerous to your system.

Please type "Steve John's Resume.doc" to open the attached file:

Opening attachments can potentially be dangerous to your system.

You are about to open "New Year Greatings.ppt.exe" which is potentially a dangerous file. Please provide the reason for doing so. Your motivation will be logged and audited.

Figure 3.3: Different types of adaptive warnings used by Keukelaere et al. [151]

As shown in the papers these measures seem to work up to some extend but also introduce major drawbacks for the user. Recording user answers and monitoring the results causes severe privacy problems and modifying the warnings that they look different each time, contradicts to the usability principle of consistency [218].

3.6.2 Guidelines and Applications Thereof

Concrete approaches in enhancing Internet warning interfaces are rare but have been done for example by Bravo-Lillo et al. [35]. They compiled five major guidelines from a large number of computer literature sources which come down to the needs to 1) following a consistent layout, 2) comprehensively describing the risk, 3) being concise, accurate and encouraging, 4) offering meaningful options and 5) presenting relevant contextual and auditing information. Using those rules they optimized existing warning messages and tested them in an online survey study. For most of their test cases understanding of the warnings did not increase significantly but they were able to measure an increase safe response rate of the participants.

The five guidelines mentioned above have their foundations in other work: Egelman et al. [78] highlight the importance of 1) interrupting the primary task, 2) providing clear choices, 3) failing safely, 4) preventing habituation and 5) altering the phishing website. The last recommendation being particularly coined for phishing research. Egelman describes more of the rules as design patterns within his PhD thesis [77]. He sees importance in 1) active warnings, 2) noticeable contextual indicators, 3) providing recommendations, 4) attractive options, 5) conveying threats and consequences, 6) considering levels of severity, 7) separating trustworthy content, 8) harden dismissing and again 9) failing safely. Besides researchers, the operating system companies also have design guidelines that to some extent address how warning design should be done [16, 26, 195]. Microsoft for example offers specific strategies of when, where and how to use error messages [193].

Cranor et al. [61] looked into design of user interfaces for privacy instead of security. As well as SSL the P3P specification describing privacy attributes is also very complex and hard to understand for the average user. Cranor et al. use a bird as a simplified representation of the matching of a privacy profile to the user needs and create a textual policy summary that should be more easily understandable by the user. For their evaluation they refer to eleven design criteria for privacy design presented by Belotti and Sellen [25]: Trustworthiness, appropriate timing, perceptibility, unobtrusiveness, minimal intrusiveness, fail safety, flexibility, low effort, meaningfulness, learnability and low cost.

Friedman et al. [97] reconsidered the design and notification process of cookies in the browser by even taking auditory feedback into account. They think that “informed consent” and “just-in-time-intervention” are key aspects for such kinds of user interfaces and created a sidebar that shows and classifies cookies on websites as they are delivered to the user’s computer. A form of “just-in-time-interventions” has also been used in one of my projects trying to protect the user from phishing websites (see section 5.5).

“Dynamic Security Skins” presented by Dhamija and Tygar [65, 66] try to establish user interface trustworthiness. Using their concept each user has a fixed randomly assigned image that appears with every login form. In addition to that they propose the idea of using a secretly exchanged hash between the server and the users commputer that is used to generate a background image for security relevant fields of the website and can also be regenerated by the user browser chrome.

Shin and Lopes [265] presented the SSL information visually in browser forms. They represent SSL status information directly in the form field either as a traffic light inside each field or by changing the background color.

Sobey et al. [270] (masters thesis: [269]) also evaluated ways of displaying the identity status (SSL-status) of a website in the browser. Using a larger area for the indicator in the browser chrome somehow enhanced the recognition of the indicator but they suggest that there have to be stronger cues for security indicators that also properly convey whether a component is interactive.

The concept of TrustBar by Herzberg and Gbara [126,127] uses a large status bar with logos to show who is identified by an SSL certificate and by whom this certificate is identified. Using a large status indicator for SSL statuses is also used in my project in section 5.6 but without wasting additional screen real estate.

WebWallet by Wu et al. [325] uses the graphical concept of separate ID cards to reveal the user's intention where she wants to submit her credentials. Immediate entry of password data is blocked by their add-on and for each website a new ID card has to be generated or an old one has to be reused. If the ID card URL does not match the website URL an unintended disclosure of user data is detected.

Wogalter et al. [318] presented a set of general guidelines for warning design that is not limited to user interfaces but should hold for every arbitrary warning. Salience (to get noticed) is one of the most important aspects for any warning. This can be achieved by high contrasts, the use of color or even special effects (like flashing). Addressing the wording of a warning four message components are essential: 1) a signal word to attract attention; 2) the possibility to identify the hazard; 3) the explanation of possible consequences and 4) information on how to avoid the hazard. Concerning the layout continuous text should be avoided in favor of bullet points for example and warnings need to be presented proximate to the hazard (time- and spacewise). Visual clutter around the warning should be reduced to make the warning stand out. Pictorial symbols can help the understanding and salience of a warning and other channels besides the visual channel may also be used were applicable (e.g. audio signals).

The wording and pictorial symbols of warning messages have been even more focused in warning research. In the study of Amer and Maris [11] the arousal strength of signal words and signal icons in warning dialogs was measured. They showed that a red cross combined with the word "critical" was the strongest possible wording and icon combination for a computer warning dialog followed by "urgent", "warning", "error" and finally "notice". Hellier et al. [121] showed that such scales remain stable no matter which other kinds of words appear in the same context.

Standardization organizations like the W3C have also tried to set up standards for security user interfaces within the world wide web. Tyler Close [53] outlines the main goals that security information in the web browser needs. He sees the consistency of terms, indicators and metaphors, as an important point as well as raising the user awareness for security information. To counteract faked information a reliable presentation of the security information is necessary whilst minimizing the number of scenarios in which the user is needed to make decisions. Close differentiates between three different use cases of security on the web: 1) providing information, 2) believing content and 3) installing software. Depending on the context of source and sender he depicts 22 different scenarios of Internet usage that can be used as scenarios to apply them against given approaches.

Markus Jakobsson [136] reports on different findings concerning the graphics and textual appearance of warnings or security related content in general. According to him, spelling and design of messages and websites are important. Emails that do not mention a concrete contact person are for example less trustworthy. Another interesting finding here was that too much emphasis on security can result in reverse effects. Very strong phishing warnings on real websites make people distrust in the website itself. For third party security logos (e.g. "'Trust-e"[24]) it is important whether the third party brand is well known. The personalization of messages is another possibility to create trust among readers. Another way is to enable content verification over different channels (e.g. by phone).

Personalization is also mentioned by Adelsbach et al. [5] whereas they refer to it in a graphical context. If users can select personalized content (e.g. a background image) a stronger relationship to the security indicator is established.

The use of a user-centered design process for security is really important because the human component of computer security has been neglected far too long [21]. Wogalter et al. [319] even considered the application of the user centered design process to the text design of classical warnings.

As important as the correct design of computer security warnings is their evaluation. Lorrie Faith Cranor [59] published a journal article about the questions that need to be asked when evaluating a security indicator or warning. Some of those being "Does the indicator behave correctly when [or when not] under attack?", "Do users know what they are supposed to do [...]?" or "Do they actually do it?".

3.7 User Study Methodology

Evaluating usable security concepts is especially hard. Security problems can only be simulated safely in the lab but this makes user studies artificial. Together with the problem that security may not be the primary task of the user it is extremely hard to create correct study methodologies for valid results. As one of the results of this thesis I have formalized the cornerstones of a standardized user study for the evaluation of anti-phishing protection in section 7.2 at the end of this thesis. Within this section of the related work chapter I want to present similar related work that gives recommendations for warning and detector evaluation in the security field.

Wogalter et al. [318] for example give a summary over the possible evaluation process (formative and summative), like design mock-up testing or testing a final warning. He recommends to measure subjective feelings of users using Likert-type scales or by using objective measures of warning effectiveness.

[24] http://www.truste.com/

Egelman et al. [79] give a good overview over the general challenges a researcher faces when doing security-related user studies. Observing users when they deal with security related issues in general is a big problem, as qualitative techniques – like interviews – do have strong limitations in this case. Stuart Schechter [257] reports on some rules on how to write up security research correctly. The reporting of the tested hypotheses, the threat model, the user study methodology and the measurements are most important to him.

Research experiments that set up their own attacks or spoof are extremely problematic, as they do not only hurt the privacy of the attacked "victims" but already by delivering an unwanted email to a user the researcher conducts a legally critical action. Because of this these types of experiments have to be checked by an IRB (institutional review board) in most countries first. Jagatic et al. [134] actually attacked users in an experiment to show the problems of socially engineered phishing attacks. This study led to large debate about such a type of attacking study [17, 55, 133].

Raffetseder et al. [242] summarized some experiences about building anti-phishing browser plugins on the more technical side. They report that building a prototype for one single browser is one thing but porting it to different browsers can get very cumbersome.

Depending on how the user study is performed other researchers have given advice for more distinct problems. Downs et al. [74] report how mechnical turk workers try to gamble research tests and how that can be avoided. Ross et al. [250] report about the average demographics of such participants, becoming increasingly international.

Other even more general related work can help in the design of usability questionnaires by using standardized questions [161], gives an overview of how to best balance within-subject conditions using Latin-squares [110] or evaluates which type of Likert-scales are best [63].

Looking at the user studies that have been carried out so far the number of real field studies is close to zero. One example from related work is the paper of Karlof et al. [145]. Within this thesis I also gathered field findings within two of my projects (see section 5.5 and section 5.6). Table 3.3 provides a good overview over different phishing user studies that have been conducted and the different dimensions that can be possibly modified. The list of possible dimensions reaches from the type of study performed (lab or field) to the number of participants, conditions and the statistical tests used for evaluation of the results. As mentioned above those dimensions are examined more closely at the end of the thesis in chapter 7.2.

Take Home Messages

➥ **3.1 The Phishing Problem:** Phishing is a severe problem causing at least millions (some reporting billions) of financial damage each year. Users fall for the attacks as security is never their primary goal. They are easily distracted by similar looking layouts and don't notice security indicators present in the browser.

First Author	Herzberg	Whalen	Dhamija	Downs	Wu	Wu	Jakobsson	Downs	Egelman	Sunshine	Kumaraguru	Sheng	Blythe	De Luca	Lin	Maurer
1) Year	**2004**	**2005**	**2006**	**2006**	**2006**	**2006**	**2007**	**2007**	**2008**	**2008**	**2010**	**2010**	**2011**	**2011**	**2011**	**2011**
2) General Evaluation/Validation of own Concept	Validation	Evaluation	Evaluation	Evaluation	Evaluation	Validation	Evaluation	Evaluation	Evaluation	Validation	Validation	Evaluation	Evaluation	Validation	Evaluation	Validation
3) Type of Study	Lab	Lab	Lab	Lab	Lab	Lab	Lab	Online	Lab	Lab	Lab	Online	Online	Lab	Lab	Lab
3a) Study Disguise	n.a.	Looking at Browsers/Webpages	none	Computer Use and Decision Making	n.r.	n.r.	none	Computer Use	Online Shopping	Usability of Information Sources	Effectively Using Emails	n.r.	none	Internet Surfing	Visual Security Aids	Internet behavior
3b) Scenario	n.a.	Role-Play	n.a.	Role-Play	Diverted Role-Play	Diverted Role-Play	n.a.	Role-Play	Real Data	Real Data	Role-Play	Role-Play	n.a.	Diverted Role-Play	n.a.	Diverted Role-Play
3c) Role Played	n.a.	Data belonging "to the lab"	n.a.	Pat/Patricia Jones	Personal Assistant	Personal Assistant	n.a.	Pat Jones	n.a.	n.a.	Bobby Smith	University Employee	n.a.	Friend of a "Good Friend"	n.a.	Grandchild of old lady
4) Study Design	Within-Subject	Within-Subject	Within-Subject	n.a.	Within-Subject	Mixed-Design	n.a.	n.a.	Between-Subject	Between-Subject	Between-Subject	Within-Subject	n.a.	Mixed-Design	Within-Subject	Mixed-Design
4a) Number of Independent Variables	1	1	1	0	3	2	0	0	1	1	1	1	0	2	1	2
4aI) Indep. Variable 1 (# of levels)	Concept used (2)	Securitybriefing (2)	Attack type (9)		Toolbar (3)	Concept Used (2)			browser/warning type (4)	Browser Type (5)	Type of Training (3)	Trained (2)		Concept Used (2)	Adressbar Hinting (2)	Concept Used (2)
4aII) Indep. Variable 2 (# of levels)					Attack Type (5)	Type of Attack (5)								Type of Data(2)		Type of Data (3)
4aIII) Indep. Variable 3 (# of levels)					Tutorial read (2)											
4b) Main Dependent Variable	Attacks detected	Indicators Looked At	Legitimate or not	Attacks detected	Attacks detected	Attacks detected	Phishiness Rating	Decision Made	Not Phished	Overrides SSL warning	Detected Phising-Mail	Attacks detected	Detected Phising-Mail	Attacks detected	Attacks detected	Attacks detected
5) False positives measured	n.r.	n.a.	yes	n.r.	yes	n.r.	n.a.	n.r.	n.a.	yes	yes	yes	yes	yes	yes	yes
6a) Overall Number of Tasks	6	5+3	20	7	20	20	26	5	2	4	19	14	20	4	32	6
6b) Number of Non-Attacking/Dummy tasks	4	2+0	8	3	15	15	n.r.	3	0	2	15	7	10	2	16	3
6c) Task to Perform	Classify Screenshot	Browse websites	Identify fraudulent websites	Handle E-Mail	Handle E-Mails	Handle E-Mails	Rate Phishiness	Handle E-Mail	Buy Something, Check Mail	Find information	Handly E-Mail	Handle E-Mail	Rate Phishiness	Perform Urgent Tasks	Website "safe"?	Perform Online-Trans.
7) Method for Balancing Tasks	none	none	randomized	sorted with increasing danger	same order	same order	n.r.	sorted with increasing danger	randomized	Reverse order	none	n.r.	n.r.	Latin-Square	randomized	Latin-Square
8) Type of Interaction	Images	Real	Real	Real	Real	Real	Real	Images	Real	Real	Real	Images	Subject and Text	Real	Real	Real
9a) Number of Participants	42	16	22	20	30	21	17	232	60	100	30	1001	224	32	22	24
9b) Age	n.r.	n.r.	18 to 56	avg 27	18 to 50	19 to 34	18 to 60	n.r.	avg 28	n.r.	avg 27	avg 30	18 to 65	21 to 27	19 to 41	avg 23
9c) Avg. Security Knowledge	high	high	high	low	avg	avg	low	high	low	n.r.	n.r.	n.r.	n.r.	avg	avg	high
10) Additional Recording Performed	none	Eyetracking	none	Video	none	none	none	none	none	Audio	none	none	none	Video	none	Video
11) Statistical Analysis Method	none	none	t, *r*	F	t	t	n.r.	r, χ^2	Fishers Exact Test	Fishers Exact Test	χ^2	F, t	F	F	F	Mann-Whitney U

χ^2	Chi-Square test	F	Analysis of Variance (ANOVA)
t	t-Test	n.a.	not applicable
r	Pearson's Correlational Coefficient	n.r.	not reported (in the paper)

Table 3.3: Comparison of different dimensions of user studies that have been conducted in research on usable security. The table has been compiled form the following references [33, 64, 67, 72, 73, 78, 126, 139, 157, 166, 183, 261, 280, 294, 324, 325]

- ➥ **3.2 The Current State of Detection Methods:** The global nature of the Internet makes law enforcement as a technique to stop phishing impossible. Today's most widely deployed anti-phishing measures are based on black- and whitelists. This technique does not only leave room for zero-hour attacks but can also be completely sidelined by clever attackers.

- ➥ **3.3 The Current State of User Intervention:** Current warnings and notifications are not noticed by the users. They are frustrated towards security and most of the time do not understand what the warning messages mean, leading to habituation and negligence.

- ➥ **3.4 Phishing Education:** Phishing education is highly controversial. Although some research proved the positive effect of training materials and that the learning effect can be retained for some time the question still remains whether people will change their behavior with better education and whether they want to invest their time in participating in training sessions.

- ➥ **3.5 Research Concepts for Detection:** All different types of technical features that are available have been used by researchers to build phishing detection methods (URLs, HTML code, CSS, images, layout, ...). In combination and with machine learning techniques they can achieve detection rates of 95% and more. But most of the approaches neglect other important factors like computation time, false positives or the possibility of phishers to adapt to the detection algorithm.

- ➥ **3.6 Research Concepts for User Intervention:** The design of new user interfaces for interventions is mostly done for arbitrary user interfaces not focusing on phishing. Using adaptive dialogs or concepts that transform security information into visual cues the compliance of the warnings can be enhanced to a small extent but mostly at the cost of usability. A lot of guidelines have been proposed that can be used by researchers and practitioners when designing new warnings.

- ➥ **3.7 User Study Methodology:** The way of evaluating detection and user intervention approaches is vital for the success of new approaches. Whatever type of evaluation is performed it is most important to carry out evaluation as close to real world security problems as possible. Keeping artificiality and security priming as low as possible contradicts with what researchers can do in their ethical and legal environment.

II

Protection through HCI

Chapter 4

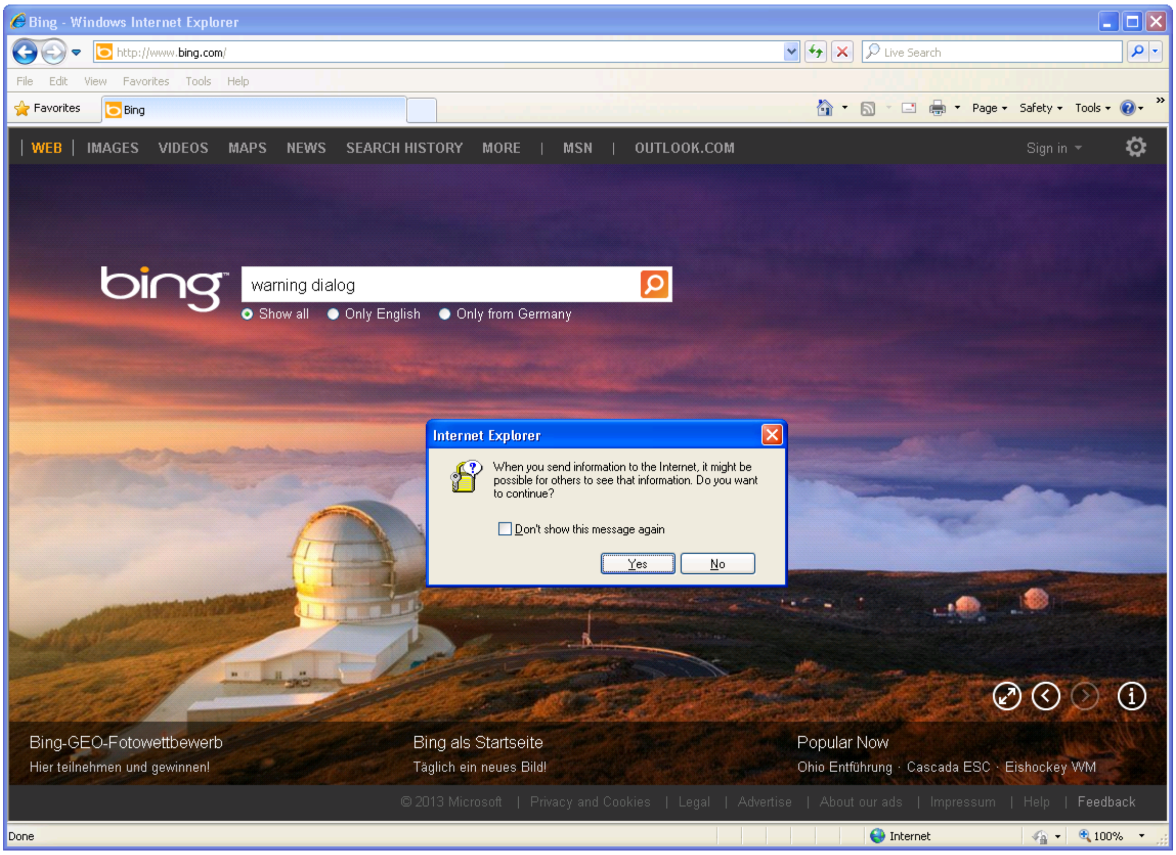
Overview of Research Covered

Before diving deeper into the research that has been conducted within the different projects that constitute this thesis (chapter 5) this short chapter gives and overview about what this thesis addresses, what contributions I make and how the research described here differs from the related work seen in chapter 3. The chapter ends with a short overview of the general structure of the remaining document.

4.1 Delimitation to Related Work

Related Work and other resources provides enough motivation and evidence that usable security and in my special case phishing is a topic well worth working on. The monetary loss through phishing attacks is – no matter how large it really is – definitely an argument for more research in this area (see section 3.1). At the moment the established countermeasures (like blacklists) are in the process of wearing off as the first phishers have found methods to completely circumvent them (see section 3.2) but phishing is not at all a solely technical problem. It is a socially engineered attack for which the user plays a very important role. Although this can already be seen from the definition of phishing in itself research in the area has mostly be on the security side of things (e.g. how to built better phishing detectors?) (see section 3.5). Usable security research has shown how and why existing methods and warnings fail (see section 3.3) but has not yet been able to come up with a proper solution for the problem.

These two sides of the coin phishing detection and user intervention are at the core of this thesis. Most of the related work done so far can be categorized belonging either to the one or the other side. Within my thesis I try to examine both parts as a whole. On the one hand



Internet Explorer

When you send information to the Internet, it might be possible for others to see that information. Do you want to continue?

□ Don't show this message again

Yes No

Bing - Windows Internet Explorer
http://www.bing.com/
Live Search
File Edit View Favorites Tools Help
Favorites
Bing
Page Safety Tools
WEB IMAGES VIDEOS MAPS NEWS SEARCH HISTORY MORE MSN OUTLOOK.COM
Sign in
bing
warning dialog
Show all Only English Only from Germany
Bing-GEO-Fotowettbewerb
Hier teilnehmen und gewinnen!
Bing als Startseite
Täglich ein neues Bild!
Popular Now
Ohio Entführung · Cascada ESC · Eishockey WM
© 2013 Microsoft | Privacy and Cookies | Legal | Advertise | About our ads | Impressum | Help | Feedback
Done
Internet
100%

Figure 4.1: A warning message of the Internet Explorer asking the user whether he wants to submit information to the Internet.

to find new ways of phishing detection derived from HCI aspects of the socially engineered attacks and on the other hand to find proper modes of user intervention mechanisms that can be interpreted by the users more correctly. With this a full understanding of the problem and solution space of phishing detection gets possible and a general idea can be generated of how approaches need to look like.

Other work not doing this could easily fall short of solving the problem. Detection concepts that lay a heavy focus on technical aspects that are easily interchangeable by an attacker will never be able to provide a future-proof protection. For example if the HTML code of websites is used for detection phishers could just change this HTML code without changing any other representation and without the user noticing. On the other hand user intervention without any properly understandable reason also falls short. The Microsoft Internet Explorer for example used to warn the user whenever information was submitted to the Internet (see figure 4.1). As a medium of information exchange submitting information to the Internet will in most cases be intended and safe.

Looking at the fields of related work previously mentioned in chapter 3 this thesis will not specifically extend the research on the amount of phishing damage does nor have I conducted specific studies to find out more about why phishing is actually successful. A small amount of new findings in this area is always generated as a matter of fact when testing anti-phishing concepts with users having a standard browser control group (e.g. the project in section 5.5). Also phishing education has not been touched explicitly within my research. As stated above the focus of this thesis lies in testing and generalizing new methods of detection of phishing attacks, better user intervention and focusing on the interplay between both.

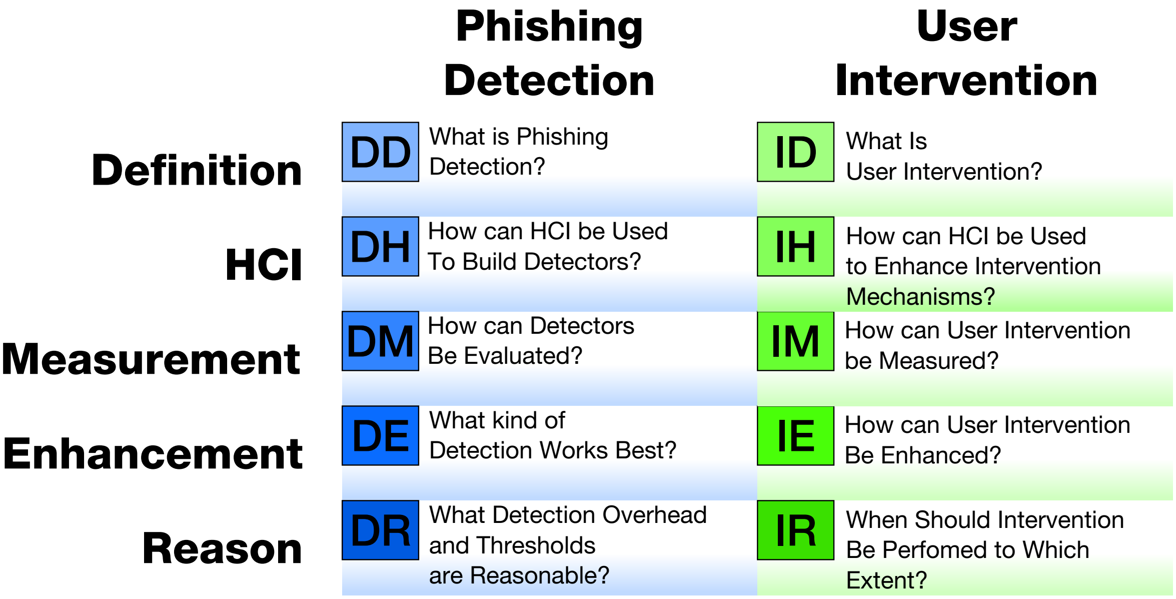
Besides the focus of this thesis being on the problem of phishing the main idea of the interplay between threat detection and user intervention is much more general problem. Most findings of this thesis should also valuable for a much broader area.

4.2 Main Research Classification

As pointed out before there are two main dimensions in the fight against phishing covered within this thesis. On the one hand the problem of **phishing detection** and on the other hand the problem of getting **user intervention** right.

For both of these dimensions we structured that research that has been carried out on five different levels:

- **Definition:** As the areas evaluated here are relatively new a clear definition of what is scope of the research and what defines the edges of the problem space is important.
- **HCI:** As both dimensions of this thesis extend existing research fields (security research and warning research) the role of HCI and how these fields can take profit of HCI is another important level of this thesis.
- **Measurement:** To be able to compare and improve different approaches it is important to know how to measure their quality. Finding such a measurements for both dimensions hence was a task of this thesis.
- **Enhancing:** The development of new concepts and methods can the be used to find answers to specific subproblems with the overarching goal to enhance the state-of-the art. The identified ways of measurement and the field of HCI are especially important.
- **Reason:** Finding a method that works well for a given problem might create a different problem somewhere else in return. As an extreme example it would suffice to get rid of the Internet as whole to get rid of 100% of all phishing attacks. Yet this is not an option. Reasoning about when and where to use which kind of method and what that might mean for other areas is a last important level.



	Phishing Detection	User Intervention
Definition	DD What is Phishing Detection?	ID What Is User Intervention?
HCI	DH How can HCI be Used To Build Detectors?	IH How can HCI be Used to Enhance Intervention Mechanisms?
Measurement	DM How can Detectors Be Evaluated?	IM How can User Intervention be Measured?
Enhancement	DE What kind of Detection Works Best?	IE How can User Intervention Be Enhanced?
Reason	DR What Detection Overhead and Thresholds are Reasonable?	IR When Should Intervention Be Perfomed to Which Extent?

Figure 4.2: Overview of the ten main research questions of this thesis being split up in two dimensions on five different levels.

4.3 Research Questions

Taking the two research dimensions and laying them out with the five different investigation levels a matrix of the ten research questions is formed. These questions will all be described within this section. Figure 4.2 gives and overview over the 10 research questions as well as providing shortcut codes for the different research questions. These will be used throughout the thesis to refer to the research questions whenever they are tackled by a project.

Phishing Detection

- **What is Phishing Detection?** Similar to the diverse range of definitions of a phishing attack or the different definition of phases of a phishing attack (see chapter 2) it is also important to define what phishing detection is.
- **How can HCI be Used to Build Detectors?** Phishing detectors have been built for a couple of years now but what are the special properties that a phishing detectors needs to have and how can they be derived using HCI and its methods?
- **How can Detectors be Evaluated?** Once a detector is built it is important to assess its quality. But how can this be done? How can results be compared to other results taken in other countries or years ago? What are important aspects that have to be part of tests?
- **What kind of Detection Works Best?** Once it is possible to measure the performance of a detector it is possible to compare different types of detectors. But what kind of

detection works best? Is is practicable? And is it future proof or can attackers adapt to it?

- **What Detection Overhead and Thresholds are Reasonable?** No matter how well a detector works it always requires a certain amount of computational effort and never has a perfect hit rate? So how good does a system need to be and how much is it allowed to intervene without disturbing too much?

User Intervention

- **What is User Intervention?** Protecting the user from phishing is also about informing the user about threats and helping him to avoid those. But what are the exact boundaries of this process of user intervention?
- **How can HCI be used to Enhance Intervention Mechanisms?** Related work shows us the many downsides current computer warnings and intervention mechanisms have. Either they go unnoticed or are annoying and lead to habituation effects. Having these findings from HCI how can new intervention mechanisms be designed to reduce those problems?
- **How can User Intervention be Measured?** What is the success of user intervention and how can this be measured. Is it enough to make sure that the user saw a user intervention or is it important that a correct action is taken? Or is the most important measure in the end that the user has been protected from any harm?
- **How to Enhance User Intervention Quality?** Taking HCI into account new user intervention methods can be developed but which are the properties of a high quality intervention method?
- **When Should Intervention be Performed to Which Extent?** Whatever type of user intervention is designed, it always interferes to some extent with the user. But to which extent should this be performed and which options should a user have to control the intervention outcome. Does anyone at all need to be able to go a phishing website? What about security researchers?

4.4 Project Overview

To be able to find answers to the research questions nine different projects have been carried out in the course of this thesis. Each single project only addressed a certain number of research questions and has its own separate findings. After the projects have been discussed one by one in chapter 5, the following chapter 6 summarizes the results and reports overall findings for the different research questions. At the beginning of each project chapter a small indicator graphic informs the reader about the different research questions tackled and at the end of each projects the research question specific findings are stated. An overview of all projects and the research questions involved can be found in figure 4.3.

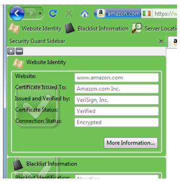
The following list gives a brief overview on each of the projects:

- **5.1 Phishing Website Test Set:** Having an extensive phishing website test base is important for getting valid results for detector testing throughout the research. This chapter introduces into the important aspects of such a pishing test set and reports the process of a test set that has been built up during this thesis.
- **5.2 SecurityGuard Website Status Rollup:** What technical properties of websites are interesting for users and how can they be offered towards them within a user intervention mechanism? Within this project we built a rating and reporting systems that displayed technical data concerning the current website within a status bar in the browser using user centered development and evaluation.
- **5.3 Community-based Rating Intervention:** In real life people often ask others for security and privacy advice even about Internet websites. Can such a concept be used online and is it as attractive to users as they make use of it offline? Within this project we built a user intervention method as a browser plugin and evaluated how such community values can effect the users' security behavior.
- **5.4 Spell Checking to Detect Fraudulent Websites:** URLs are an important indicator for detecting phishing attacks as they cannot be as easily impersonated like the website contents of a company. With proper knowledge users could easily find a lot of attacks when investigating the URL. Because of this many phishers try to create URLs that look similar to a trustworthy company URL. Can this behavior of the attackers be used for detecting phishing attacks in an automated manner. In this subchapter we present a detector and its evaluation that is based on this fact.
- **5.5 Data Type Based Security Dialogs:** Security warnings are all around the computer and the Internet browsers but when do critical security decisions happen? In this subchapter we take the advantage of the fact that phishing only happens if critical types of data (e.g. credit card numbers) are involved. This can be used for filtering incoming attacks and allows to create a user intervention method that takes the user's context into account. The development of this user intervention method and an extensive evaluation is presented within the respective subchapter.
- **5.6 Enhancing SSL Awareness in Web Browsers:** Non-blocking indicators are usually said to remain unnoticed by the users. This has been proven for lock-icons and other smaller security indicators. Within this project we want to test this again by using a large area of the whole background of the browser user interface to report the SSL status. We evaluate this concept by looking at whether our plugin is able to change the user's attitude towards websites in dependence of the notification shown.

5.1 Phishing Website Test Set

Having an extensive phishing website test base is important for getting valid results for detector testing throughout the research. This chapter introduces into the important aspects of such a pishing test set and reports the process of a test set that has been built up during this thesis.

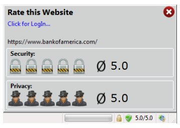
DD	ID
DH	IH
DM	IM
DE	IE
DR	IR



5.2 SecurityGuard Website Status Rollup

What technical properties of websites are interesting for users and how can they be offered towards them within a user intervention mechanism? Within this project we built a rating and reporting systems that displayed technical data concerning the current website within a status bar in the browser.

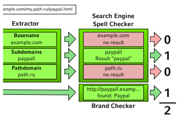
DD	ID
DH	**IH**
DM	**IM**
DE	**IE**
DR	IR



5.3 Community-based Rating Intervention

In real life people often ask others for security and privacy advice even about Internet websites. Can such a concept be used online and is it more attractive to users as they make use of it offline? Within this project we built a user intervention method as a browser plugin and evaluated possible effects.

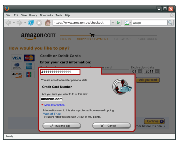
DH	ID
DH	**IH**
DM	**IM**
DE	**IE**
DR	**IR**



5.4 Spell Checking to Detect Fraudulent Websites

URLs are an important indicator for detecting phishing attacks as they cannot be as easily impersonated. Can similar looking URLs be used to detect phishing attacks automatically? In this chapter we present a detector and its evaluation that is based on URL similarity.

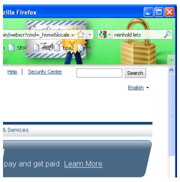
DD	ID
DH	IH
DM	IM
DE	IE
DR	IR



5.5 Data Type Based Security Dialogs

In this subchapter we take the advantage of the fact that phishing only happens if critical types of data (e.g. credit card numbers) are involved. This can be used for filtering incoming attacks and allows to create a user intervention method that takes the users' context into account.

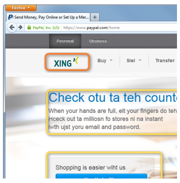
DD	ID
DH	**IH**
DM	**IM**
DE	**IE**
DR	**IR**



5.6 Enhancing SSL Awareness in Web Browsers

Non-blocking indicators are usually said to remain unnoticed by the users. This has been proven for lock icons and other smaller security indicators. Within this project we test this again by using the whole background of the browser user interface to report the SSL status.

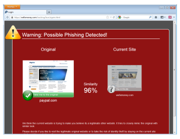
DH	ID
DH	**IH**
DM	**IM**
DE	**IE**
DR	**IR**



5.7 Diminishing Visual Brand Trust

The content area of the web browser is the most important place for users to assess the trustworthiness of a website although it is easily impersonated. In case of this project we looked at whether small website content changes lead to a stronger focus on other security indicators in the browser.

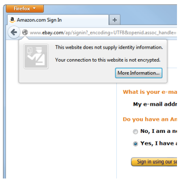
DH	**ID**
DH	**IH**
DM	**IM**
DE	**IE**
DR	IR



5.8 Visual Image Comparison For Phishing Detection

The visual similarity of impersonating websites towards their original websites makes users often trust into these faked websites. We built a detector for phishing websites using visual similarity and tested different image features as well as a user intervention method for this kind of detection.

DD	ID
DH	**IH**
DM	**IM**
DE	**IE**
DR	IR



5.9 The User Study Web Browser

Using live original and phishing websites in user studies brings along a variety of problems in a study setup that needs to fulfill certain parameters. Within this project we built a browser plugin that makes it possible to mimic arbitrary security situations and finally tested whether those changes can be detect.

DH	ID
DH	IH
DM	**IM**
DE	IE
DR	IR

Figure 4.3: Overview of all projects carried out throughout this thesis each showing which research questions have been covered.

- **5.7 Diminishing Visual Brand Trust:** The content area of the web browser is the most important place for users to assess the trustworthiness of a website. However, it is also one of the regions that can be most easily modified by an attacker. Users often report that they get suspicious towards email messages in case of incorrect spelling for example or weird graphics. This is what we apply within this project. In case a website content is changed to confuse the user viewing it, does this make her rely more on other security indicators outside of the content area?

- **5.8 Visual Image Comparison For Phishing Detection and Reporting:** The visual similarity of impersonating websites towards their original websites makes users often trust into these faked websites. This is the reason why attackers try to make their phishing attempts look very similar to the original websites the users are used too. But image similarity and comparison is a technically advanced field. This is why we tried to build a detector that tries to detect phishing websites through visual similarity. Within the project we tested different image features that can be used for comparison and also developed and evaluated a user intervention method for this kind of detection.

- **5.9 The User Study Web Browser:** In user studies focusing on phishing it is nearly always necessary to confront the participants with phishing and original websites. Using real phishing websites brings along a variety of problems and even using the real non-malicious websites can be hard in a study setup that needs to fulfill certain parameters. This is why experimenters usually rebuilt both kinds of websites for user studies. A problem that arises here is to easily fake domain names and especially security indicators that are bound to network properties that cannot be easily influenced. Within this project we built a browser plugin that makes it possible solve all these problems and finally tested whether its changes can be detected by study participants or not.

Chapter 5

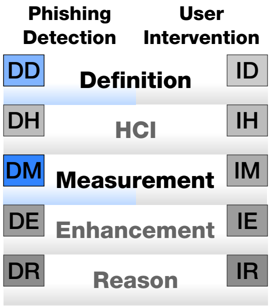
Nine Research Projects on Phishing and Usability

To gather new findings and evaluate the connections between good phishing detection and good user intervention to achieve a final perfect protection this long chapter contains nine different research projects that all focus on different parts of the research questions. Each subchapter will contain motivation for the respective project, and explanation of the general concept involved and the evaluation and results gathered within this project. In the end of each chapter the findings that can be applied towards each of the tackled research questions are summarized.

Throughout this chapter I will always use “we” as the grammatical person in describing all work whether it was carried out by me alone or with the help of other researchers or students. Where applicable I will give credit to all persons involved at the beginning of each subchapter. Whenever German has been used as a language in user studies or surveys I will rephrase the answers and questions to the closest English equivalent without separately stating that German was the original language.

Within the subchapters I will not state any hypothesis that we had while the projects were carried out but still describe our study methodology, dependent and independent variables and results towards the central topics of each chapters. As the stated hypotheses were in many cases obvious and are easily deductible from the given independent variables we did not feel the need to include them. Another reason why we will not report the hypotheses is that many interesting research findings in our projects have been made aside from the general hypotheses that had been formulated.

5.1 Phishing Website Test Set



Within this subchapter we present a project in which adequate website tests for phishing research were identified. We describe the process of building up a phishing test set, available for testing in other projects of this thesis and as a comparable basis for other researchers. In the following subsections we will first give an introduction on how a phishing test set should look like reasoning about which parameters are important for a test set to cover important phishing aspects a detector should be tested for. Afterwards we describe the collection sources and collection procedure we used for our test set. To suit all requirements defined for our test set a separate manual post processing phase we needed that was performed on the collected data. With the complete test set it was possible to retrieve accumulated findings from it and report about the possible application areas of the test set. The subchapter concludes with the research results this project contributes to the overall research findings.

5.1.1 What Should a Phishing Test Set Look Like?

The evaluation of heuristic detectors for any property is usually done by determining the qualitative performance of a given detector in terms of false positives, false negatives and their “true” counterparts (see section 1.4). In case of phishing, true positives are the phishing websites successfully detected; false negatives are situations whenever the detector failed to detect a phishing website. To cover these two detector measurements in an evaluation the test set must contain phishing websites. True negatives and false positives are the counterparts referring to non-phishing – or original – websites. To test these properties of a detector non-phishing websites must be part of the test set. Without the coverage of both aspects a detector that would simply accuse every website of being phishing would have a perfect detection rate for phishing websites. The inability of classifying original websites as well would go unnoticed. Summing this up it is of utmost importance that a test set for phishing websites needs to cover both phishing and non-phishing websites to some degree.

Another important point that can optionally be taken into account is the matching between phishing and non-phishing websites. “Matching” in this case means that for each phishing website that impersonates another website the original website is also part of the test set. A test set that contains a number of arbitrary phishing and non-phishing websites that have nothing in common might be bad for some kinds of detectors. As most phishers try to impersonate existing websites it should be made sure that detectors can detect the difference between the true original website and the phishing website. Having such a kind of test set it gets even possible to verify whether the detector is able to find the correct original “parent” website to a given phishing website. To be able to test this, a linkage between each phishing website and their original counterpart is desirable.

For the evaluation of some of the detectors in this thesis we collected test data according to the previously named characteristics. The collected test set was not only usable for testing but made it also possible to gather some findings about the current state-of-the art of phishing attacks and the current phishing landscape. Before describing the collected parameters in more detail in section 5.1.2 we will start by describing the overall collection procedure. In the end our test set was made publicly available for download to other researchers[1].

5.1.2 Collection Phase

The collection phase of the phishing test set consisted of three separate steps. In a first step possible sources of phishing and non-phishing websites had to be identified before setting up the parameters of the collection process. Afterwards the main collection procedure – using our own crawling software described in section 5.1.2 – gathered data from the sources and collected all the different parameters for each website.

Collection Sources

The largest public and up-to-date source for existing phishing attacks is phishtank.com[2], a website dedicated to the collection of possible phishing attacks which are then manually verified by a community. A list of the verified phishing URLs can be publicly downloaded containing the URLs of the websites together with some other metadata: a unique PhishTank id, the online status, the time of submission and the time of verification. Additionally some domain specific properties fetched by PhishTank like the IP-address of the server are reported. A “target” parameter is denoted to contain the targeted brand of the attack but in most cases this parameter field only contains the string “Other”. Other parameters like the HTML code of the website or screenshots are not available although phishtank.com seems to collect them as they display a very low quality screenshot with a watermark on the detail pages for every phishing attack.

phishtank.com collects vast amounts of phishing URLs. In January 2013 for example they identified 25,021 valid phishes from a number of 37,811 submissions of potential phishing URLs [234]. The median time for the manual verification in this month was 3 hours and 31 minutes.

As an initial basis for our scans we collected 10,030 phishtank.com URLs that were reported to be still online. They had been submitted to phishtank.com around the 20th march of 2012. We stored all URLs and metadata available from phistank.com in a local database for further processing.

Regarding the non-phishing websites we used alexa.com, a website of a company dedicated to the collection of global website analytics [9]. These global website traffic estimates and

[1] http://www.medien.ifi.lmu.de/team/max.maurer/files/phishload/

[2] www.phishtank.com

other metrics are calculated through data that is collected from toolbar users. The company states that they have “millions of worldwide internet users”.

Alexa offers a list of the top 1,000,000 websites for free download. As most of the phishing attacks that impersonate a specific company assume that the recipient of the attack is a customer of the respective brand phishers choose popular websites. We chose to include the 1,000 most popular URLs of this list into our index for non-phishing websites. The list only contained a numbered index followed by the URL of the website server – no other metadata. Using the first 1,000 websites of the list still did not guarantee that we would have covered the original website to every phishing attack. Because of this we added missing original websites in the final classification phase and recollected their data.

Collection Parameters

Before starting with the actual collection phase for the websites the parameters that should be captured during collection had to be defined. So far we had only stored the phishing and non-phishing URLs in our database together with the little meta information that we had from our sources.

For our collection procedure we then wanted to gather or compute the following additional information:

- **URL hash:** As URLs can be long a MD5-hashed version of each URL was computed to make it possible to quickly make sure that a URL was unique in our database.
- **Final URL:** When visiting the URLs from the list it is necessary to follow existing redirects until the browser reaches the actual website that would be displayed in a user’s web browser. This URL may be different from the URL that was used to start the request.
- **URL/Final URL basedomain:** We also extracted the basedomains from the start and the final URL and stored them separately.
- **Status Code:** For every request that was made, the HTTP status code[3] was noted. This made it possible to quickly find websites that were offline or stopped working by just examining their HTTP status code. Nevertheless these checks are not perfectly reliable. A warning message of a webhoster saying that a web space was closed may for example be served with a status code of 200 (OK) and the other way round a phishing site could conceal itself as being the error page for a 404 (Not found) status code.

[3] HTTP status codes are numeric return values for a HTTP request that express the success or failure of a request to a HTTP server. The most important status codes for example are: 200 “OK”, 404 “Not found”, 500 “Internal Server Error”, 301 “Moved Permanently” [88, 132].

- **HTML Content:** The returned HTML source code was also stored but only the very first level. This means that no images, stylesheets or other referenced data was downloaded and we also did not follow framesets or iframe[4] references that would load other websites as subwebsites of the current website.
- **Load Time:** Depending on the server connection with the respective web server the times for retrieving the website and its information varied. Hence the time from beginning the HTTP request until the website was completely loaded was also measured and stored.
- **Scanning Timestamp:** The point in time of completing the loading of the URL and additional information was stored for each website to store a record when the respective data was exactly fetched.
- **Screenshots:** Besides the HTML contents of the requested website we also acquired three different types of screenshots: 1) a browser screenshot of the web browser window together with the URL bar and all other information; 2) a cropped screenshot showing only the visible contents of the website and 3) a full page screenshot capturing the whole contents of the loaded website no matter how long the website was. Type 3 hence included all elements that would only be visible by scrolling down within the active window.

Collection Procedure

The data collection was performed in an automatic manner using two different techniques within a Java software. To fetch the website contents a HTTP request was executed and the return values were stored. For capturing screenshots it was necessary to load the website including all stylesheet and image information and then render it as it would have been done by a standard webbrowser. For this task we used an off-the-shelf Firefox browser that was remotely controlled by the Selenium web browser automation framework. Selenium is used to automate any kind of browser interaction (usually for web application testing purposes)[5]. With the Selenium WebDriver project[6] it is possible to remotely carry out browser actions a user would perform from the foundations of a programming language.

We used a standard office computer (multi-core Windows 7) for collecting the data of the different websites. For security reasons of the host system some of the scans have been done within a virtual machine hosting Windows XP.

[4] The frameset and iframe-Tags of HTML allow to embed other websites within one existing website. A frameset can be used to compile a website consisting of several websites by splitting the screen whereas an iframe allows the placement of an external website in a separate container [289].

[5] `http://docs.seleniumhq.org`

[6] `http://docs.seleniumhq.org/projects/webdriver/`

5.1.3 Post Processing

After fetching all additional data of the website a manual post processing phase was needed to complete the test set. On the one hand this was needed due to the requirement that for each phishing website the linkage to the respective original website should be determined. On the other hand this was necessary because not every visit to a phishing website necessarily led to the HTML code and a screenshot of phishing website. Although the data collected from phishtank.com stated that all phishing URLs were still online a lot of the requested websites were already taken offline or replaced by error pages. A reason for this is the high fluctuation that phishing websites have. The timeframe between acquiring online URLs and the moment that they were actually scanned by our framework could in many cases be so large that the phishing website already went offline again. This shows another reason why a working "offline" phishing test set is so important for testing detectors. With a consistent non-changing test set it can be guaranteed that multiple test runs with the same detector would lead to the same result. Testing against live phishing websites would make it impossible to produce constant and comparable results.

Website Classification

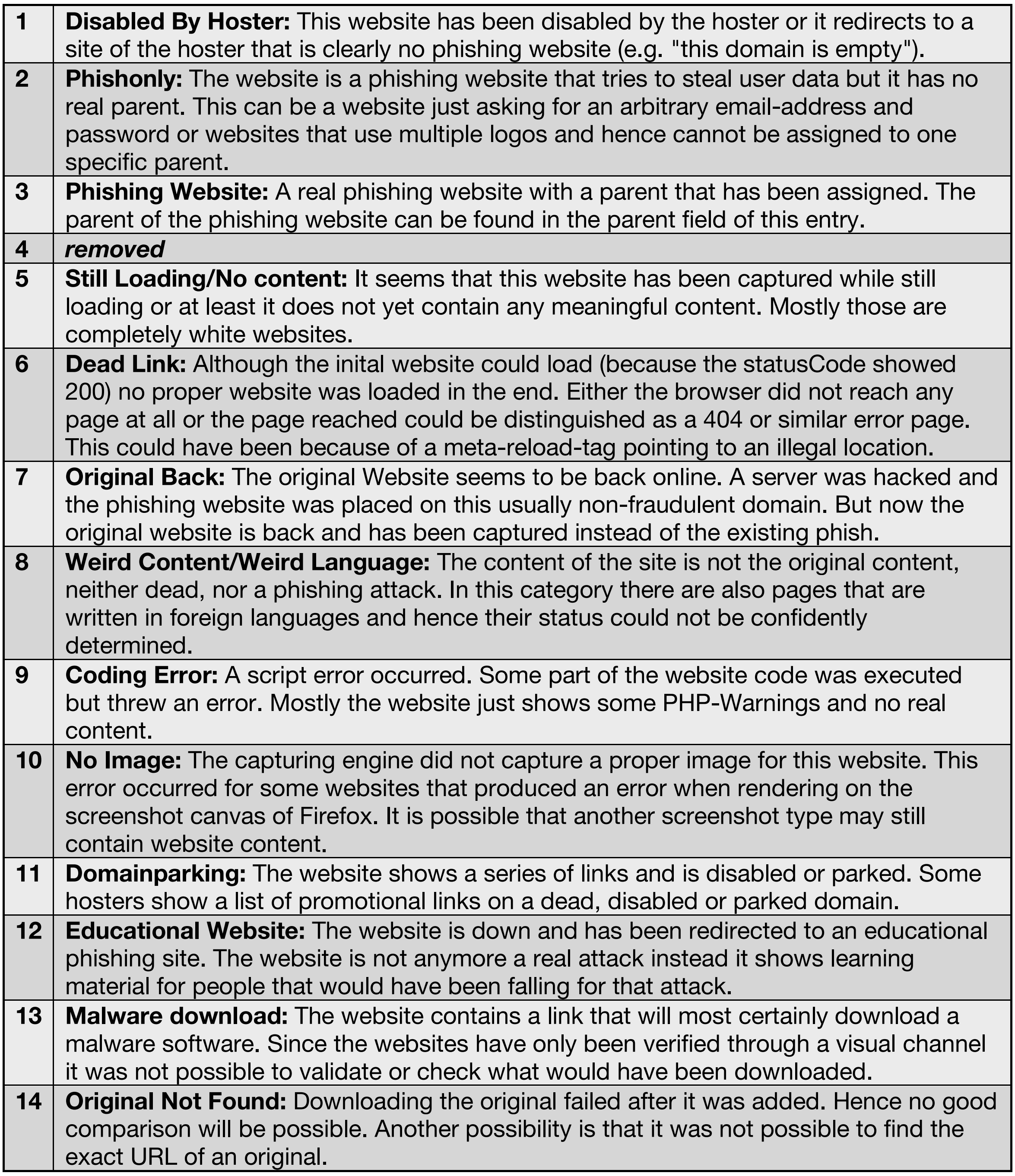
The analysis of the retrieved websites showed that there is quite a variety of cases that can be observed. A classification of these is given in table 5.1.

In case real phishing websites have been captured, one has to distinguish between phishing websites that mimic one single parent website (state 3) – the ones that we were actually looking for – but there are also phishing websites that use multiple brand logos or no brand logo at all to fetch arbitrary email credentials (state 2).

All other states relate to websites that did not load an online phishing attack. A large number of those are websites that simply did not load because they were already offline (state 6), but other cases like coding errors (state 8) or even educational anti-phishing explanations that have been put up on server of the former phish (state 11) were present. The assignment of those states to the websites has been done during the website linkage phase explained below.

Website Linkage

Linking phishing websites to their respective original websites has to be done manually as no perfect detection methodology for such a case exists – this is the reason why the test set is needed at all. To process the 10,030 captured phishing websites we hence wrote a small web user interface that was used to assign each website to its respective parent. Using a three step process each of the phishing websites was classified to the respective state and if necessary was assigned the respective original website.



1	**Disabled By Hoster:** This website has been disabled by the hoster or it redirects to a site of the hoster that is clearly no phishing website (e.g. "this domain is empty").
2	**Phishonly:** The website is a phishing website that tries to steal user data but it has no real parent. This can be a website just asking for an arbitrary email-address and password or websites that use multiple logos and hence cannot be assigned to one specific parent.
3	**Phishing Website:** A real phishing website with a parent that has been assigned. The parent of the phishing website can be found in the parent field of this entry.
4	***removed***
5	**Still Loading/No content:** It seems that this website has been captured while still loading or at least it does not yet contain any meaningful content. Mostly those are completely white websites.
6	**Dead Link:** Although the inital website could load (because the statusCode showed 200) no proper website was loaded in the end. Either the browser did not reach any page at all or the page reached could be distinguished as a 404 or similar error page. This could have been because of a meta-reload-tag pointing to an illegal location.
7	**Original Back:** The original Website seems to be back online. A server was hacked and the phishing website was placed on this usually non-fraudulent domain. But now the original website is back and has been captured instead of the existing phish.
8	**Weird Content/Weird Language:** The content of the site is not the original content, neither dead, nor a phishing attack. In this category there are also pages that are written in foreign languages and hence their status could not be confidently determined.
9	**Coding Error:** A script error occurred. Some part of the website code was executed but threw an error. Mostly the website just shows some PHP-Warnings and no real content.
10	**No Image:** The capturing engine did not capture a proper image for this website. This error occurred for some websites that produced an error when rendering on the screenshot canvas of Firefox. It is possible that another screenshot type may still contain website content.
11	**Domainparking:** The website shows a series of links and is disabled or parked. Some hosters show a list of promotional links on a dead, disabled or parked domain.
12	**Educational Website:** The website is down and has been redirected to an educational phishing site. The website is not anymore a real attack instead it shows learning material for people that would have been falling for that attack.
13	**Malware download:** The website contains a link that will most certainly download a malware software. Since the websites have only been verified through a visual channel it was not possible to validate or check what would have been downloaded.
14	**Original Not Found:** Downloading the original failed after it was added. Hence no good comparison will be possible. Another possibility is that it was not possible to find the exact URL of an original.

Table 5.1: List and description of the different states that have been assigned to the phishing websites fetched for the phishing website test set.

Original Websites and Brands

Although one might think that it is enough to only assign one parent website to each phishing website this concept is problematic. For example, each original website may have numerous domains that are to a certain extent all very similar (e.g. google.com, google.co.uk, google.de). If a Google phishing website is not classified as being a fake of google.com and a detector would match it to google.co.uk this could be interpreted as an incorrect match. However, in a real life situation there is a meta-level to this problem: multiple original websites can belong to the same brand. Detecting that a phishing website impersonates any of those brand websites would already be a valuable detection result. To make it possible to detect this, we introduced the concept of brands into our data model creating a brand for each original website and assigning other original website to the same brand were needed.

If a detector would now detect a phishing website of google.com as being a phishing website of google.co.uk one could still see that both original websites belong to the brand of "Google" and hence the detection result would still be okay.

5.1.4 The Final Test Set

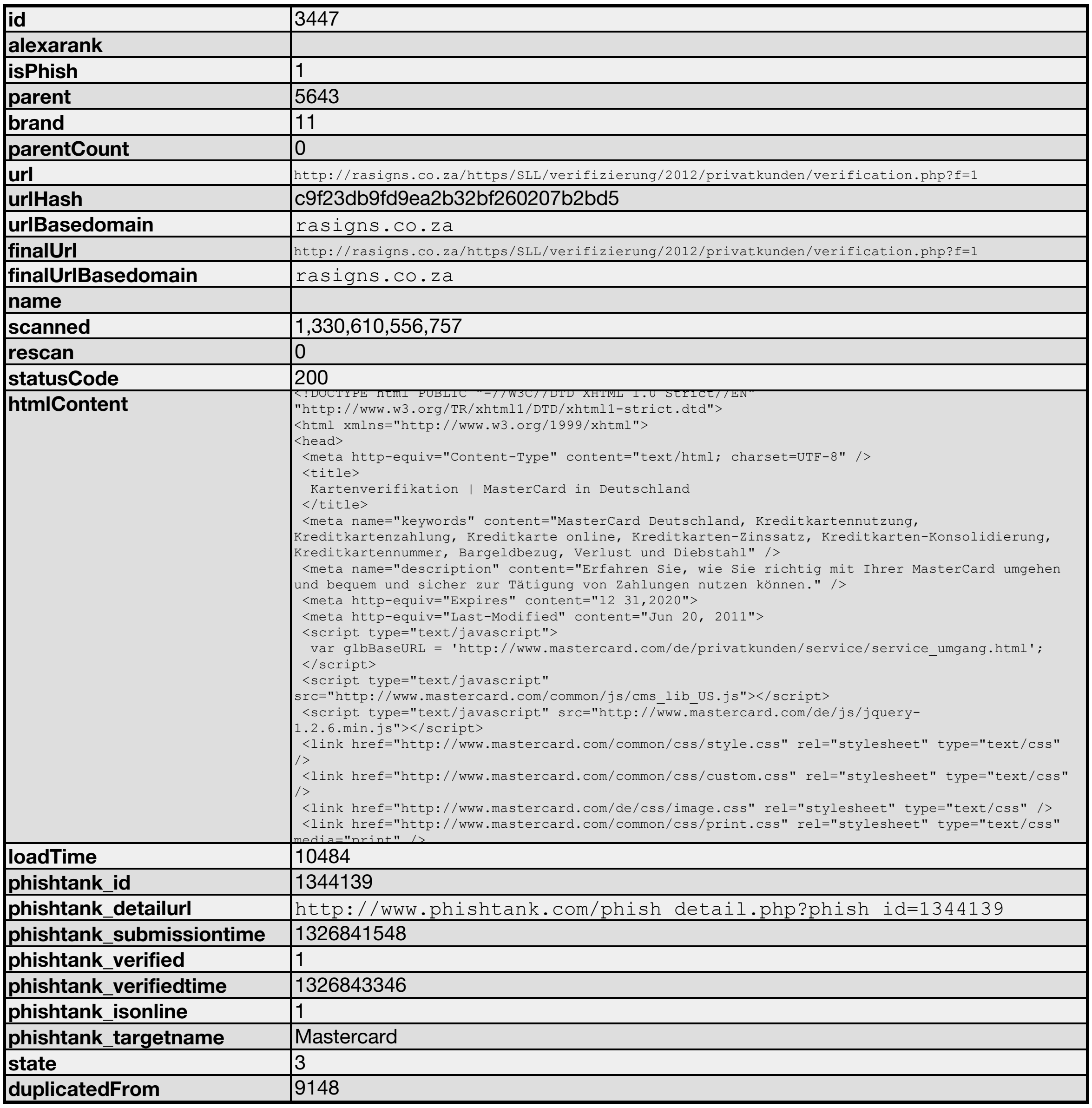
For the final test set that we made public we chose to keep all test data. For some researchers the remaining states might also be valuable in some cases. If not they could easily query the subset of attacks of state 2 and possibly state 3. After adding the new original websites where necessary in the classification phase, our final database had a total of 11,182 URLs of which 10,030 were phishing URLs and 1,152 were original website URLs. Each website has a database record similar to the example record in table 5.2.

5.1.5 Findings from of the Test Set Data

After the classification phase was finished the test set itself was a valuable source for interesting findings.

Concerning the original URLs we assigned those to 1,097 unique brands. Most brands so far only consisted of one single URL but for 20 brands two or more URLs were assigned. Google being the number one brand with 69 URLs, followed by eBay with six different URLs, the TAM Airline with 4 different URLs and Microsoft, the Pirate Bay and Orkut with 3 different URLs. Some Brazilian services had a high frequency of phishing attacks in our test set (see below) explaining the TAM Airline turning up in this list.

Looking at the classification of the 10,030 phishing URLs, real phishing websites were still the largest part with 3,603 (43%) correctly downloaded phishing attacks that we could assign to an original website parent. Still a total of nearly 60% of our downloaded websites did not turn out as being working online attacks and belonged to other categories. 21% of the websites were already disabled by the hosting company and 15% of the URLs were



id	3447
alexarank	
isPhish	1
parent	5643
brand	11
parentCount	0
url	http://rasigns.co.za/https/SLL/verifizierung/2012/privatkunden/verification.php?f=1
urlHash	c9f23db9fd9ea2b32bf260207b2bd5
urlBasedomain	rasigns.co.za
finalUrl	http://rasigns.co.za/https/SLL/verifizierung/2012/privatkunden/verification.php?f=1
finalUrlBasedomain	rasigns.co.za
name	
scanned	1,330,610,556,757
rescan	0
statusCode	200
htmlContent	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> <html xmlns="http://www.w3.org/1999/xhtml"> <head> <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" /> <title> Kartenverifikation | MasterCard in Deutschland </title> <meta name="keywords" content="MasterCard Deutschland, Kreditkartennutzung, Kreditkartenzahlung, Kreditkarte online, Kreditkarten-Zinssatz, Kreditkarten-Konsolidierung, Kreditkartennummer, Bargeldbezug, Verlust und Diebstahl" /> <meta name="description" content="Erfahren Sie, wie Sie richtig mit Ihrer MasterCard umgehen und bequem und sicher zur Tätigung von Zahlungen nutzen können." /> <meta http-equiv="Expires" content="12 31,2020"> <meta http-equiv="Last-Modified" content="Jun 20, 2011"> <script type="text/javascript"> var glbBaseURL = 'http://www.mastercard.com/de/privatkunden/service/service_umgang.html'; </script> <script type="text/javascript" src="http://www.mastercard.com/common/js/cms_lib_US.js"></script> <script type="text/javascript" src="http://www.mastercard.com/de/js/jquery-1.2.6.min.js"></script> <link href="http://www.mastercard.com/common/css/style.css" rel="stylesheet" type="text/css" /> <link href="http://www.mastercard.com/common/css/custom.css" rel="stylesheet" type="text/css" /> <link href="http://www.mastercard.com/de/css/image.css" rel="stylesheet" type="text/css" /> <link href="http://www.mastercard.com/common/css/print.css" rel="stylesheet" type="text/css" media="print" />
loadTime	10484
phishtank_id	1344139
phishtank_detailurl	http://www.phishtank.com/phish detail.php?phish id=1344139
phishtank_submissiontime	1326841548
phishtank_verified	1
phishtank_verifiedtime	1326843346
phishtank_isonline	1
phishtank_targetname	Mastercard
state	3
duplicatedFrom	9148

Table 5.2: An example record of one phishing website from our test set database. Besides this information the three linked website screenshots were stored in a different database table with links to the parent website.

ID	Count			Category Name
3	3603		🟦	Phishing Website
1	1743		🟥	Disabled By Hoster
6	1294		🟩	Dead Link
2	348		🟪	Phishonly
7	299		🟦	Original Back
11	280		🟧	Domainparking
9	212		🟦	Coding Error
8	178		🟥	Weird Content/Weird language
5	163		🟩	Still Loading/No content
12	121		🟪	Educational Website
14	107		🟦	Original Not Found
13	22		🟧	Malware download
10	19		🟪	No Image

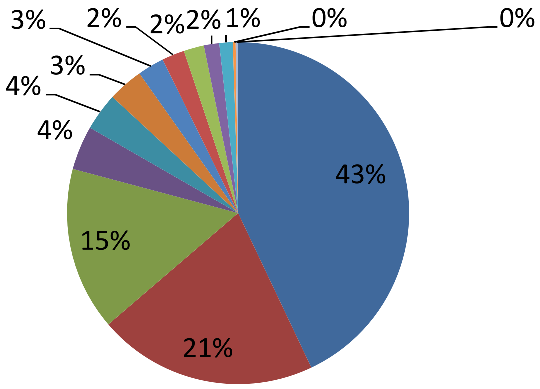


Figure 5.1: Distribution of the different phishing website states over the 10,030 phishing URLs that have been scanned.

completely dead. One of the most interesting findings of this list is the rather high number of "Phishonly" attacks. About 4% of all the phishing URLs did not target a specific brand but instead simply asked for arbitrary credentials or used multiple company logos. Compared to the 3,603 other phishing attacks found this is even 8,8% of all online phishing attacks. This clearly shows that no matter how successful such attacks are they are used by some phishers and need to be regarded as possible threats as well. For a detailed list of all states please refer to figure 5.1.

As reported above we had a list of 1,152 non-phishing URLs that were reduced to 1,097 unique brands. But how many brands of those have been phished and which ones are the most attacked ones? When measuring this data we did not only include all websites of state 3 ("Phishing Website") but also the phishing websites of state 14 ("Original Not Found"): although we were not able to find the original website that had been faked for some of the attacks it was still possible to assign a unique brand to those attacks. In total, we found at least one attack for 208 brands out of the 1,097 original brands (28%) that we had in our database. The top 13 of those brands – mentioned in figure 5.2 – already cover 78% of all the attacks that were analyzed. By far the brand that was most present in our test set was PayPal with 36% of all attacks accounting for phishes relating to them. PayPal is followed by the online community platform Habbo (9%), the social network Orkut (6%) and the auction platform eBay (5%).

Looking at this one can clearly see a huge gap between PayPal being the most phished brand and the other brands in the list. In fact there is another explanation for that besides the fact that PayPal is by far mostly attacked. Laura Oppenheimer from OpenDNS [224] reports that PayPal actively pushes phishing attacks reported to them to PhishTank (our data source) and so far seems to be only company doing that. This may account for the skewed data showing a lot more phishing attempts to PayPal than to any other brand. In fact it is hard to assume how many PayPal phishes would have been reported without the help of the company itself but the number would most certainly be lower than it actually is. Applying this to the rest of the

Count			Brand Name
1316		🟦	Paypal
335		🟥	Habbo
233		🟩	Orkut
197		🟪	eBay
190		🟦	TAM Airlines
120		🟧	Santander
88		🟦	Cielo Lottery
88		🟥	AOL
79		🟩	Tibia
70		🟪	Battle.net
57		🟦	Mastercard
56		🟧	Visa
52		🟦	RuneScape
829		🟥	195 other brands
3710			**Total**

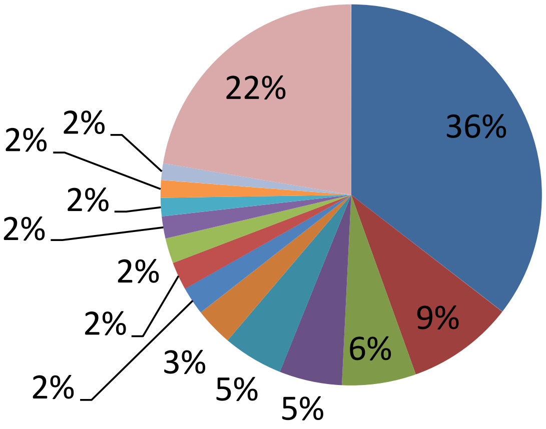


Figure 5.2: The top 13 of the 208 brands phished in total and the number of occurrences in the test set. We included all websites of state 3 and 14.

data would hence mean that a lot of actual attacks are missed and not reported to PhishTank at all. Assuming that PayPal would have the same number of reported phishing attempts as the second most prominent brand in our dataset (355 found phishing websites instead of 1316) this would mean that only 27% of all phishing attacks of a brand are actually reported to PhishTank and that the actual phishing numbers are more than three times higher than actually assumed. As interesting as those calculations are, they are also highly speculative as the real number of PayPal reports that would occur without their own reporting cannot be determined and if it was even possible this would necessarily mean that the difference is the same for all other brands.

When using a test set such as the one presented here for evaluation of phishing detectors, another property of phishing attacks is highly important: the distribution of attacks across the set. The order of attacks in our test set corresponds to the the time of reporting of an attack. As phishers usually not only launch one single attack for a brand but spread it over a larger number of URLs, the timeframe – and thus the number of different attacks – has to be carefully selected. Even for high frequency attacks like PayPal this problem persists. As one can see in figure 5.3, the position of PayPal attacks throughout the test set varies (see chart 5.3a). Parts (b) and (c) of the figure show a subset with the first 100 and 500 PayPal phishing attacks of our test set. Although the average percentage in all three cases is approximately 35% this is clearly not the case for all every portion of the test set. 100 attacks from the offset of attack 1,000 have only 27% phishing attacks (see 5.3d). Looking at (a) again, one can see that there would be even less attacks in the area from 1,200 to 1,800 and a lot more attacks at around 2,900. In practice this means that phishing tests with real life

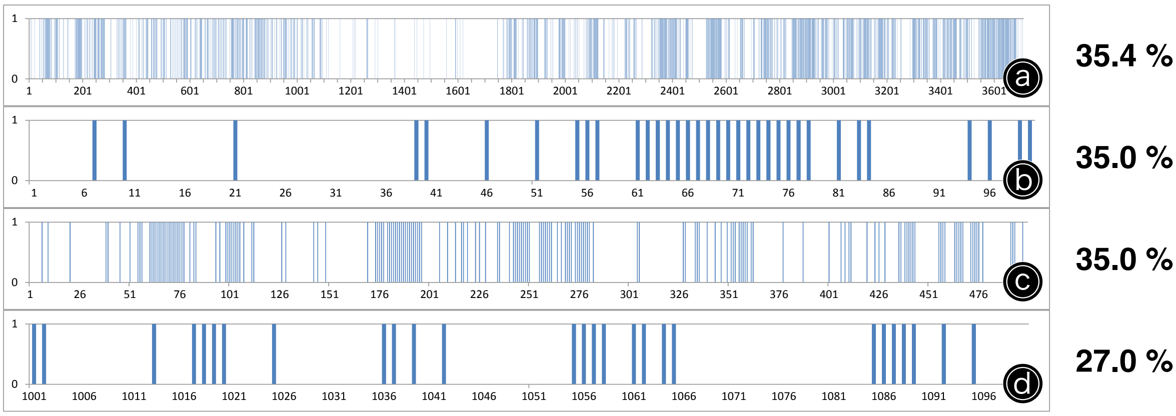


Figure 5.3: Grouping effects that occur in smaller sized phishing test sets. This figure looks at PayPal phishing attacks and their distribution throughout different portions of our phishing test set (ordered by the time of addition to our database). a) PayPal attack distribution among all 3,710 attacks; b) PayPal distribution among first 100 attacks; c) PayPal distribution among first 500 attacks; d) PayPal distribution among 100 attacks starting from offset 1001. The distribution of the same attack variestion which may lead to seemingly different attack percentages.

test sets should always contain a large number of tests as smaller numbers might be biased by the current wave of attacks being launched by phishers.

Many of the phishing attacks try to closely mimic their parent website as close as possible, but for varying reasons other not so perfectly looking ones do also exist. Figure 5.4 shows eight different PayPal phishing website screenshots from our test set. Although many of the screenshots look very close to one of the real PayPal phishing websites that existed at that time (e.g. ID 1117 and 1179) others fail to look exactly the same. In some cases this is due to technical problems or the inability of the phisher to produce a similarly looking result (ID 3234). In other cases these can also be hosting limitations of the webspace chosen by the phisher (ID 3413 and 3533; e.g. a blog hoster that allows only to place content in a predefined layout). Some phishers deliberately create a different layout to place a web form gathering data within this layout (ID 4624 and 2714). ID 7846 shows a very interesting phishing attempt that tried to recreate the look and feel of the standard website but used a completely different photo and claim taken from a modified stock photography image.

Some other interesting screenshot examples to look at are the 8.8% of phishing websites that did not target a specific brand. Looking at the examples in figure 5.5 one can see that most of them are used to gather email credentials (all except ID 1458). Since these credentials usually contain the email provider within the username no specific brand setup is needed. Some of them are done using online service that allow to setup online forms (e.g. Google Docs[7]) (ID 1005). Others place a multitude of different brand logos aside a login form (ID 1275 and 7428). ID 1300 used the same technique but here additionally the brand colors of

[7] http://docs.google.com

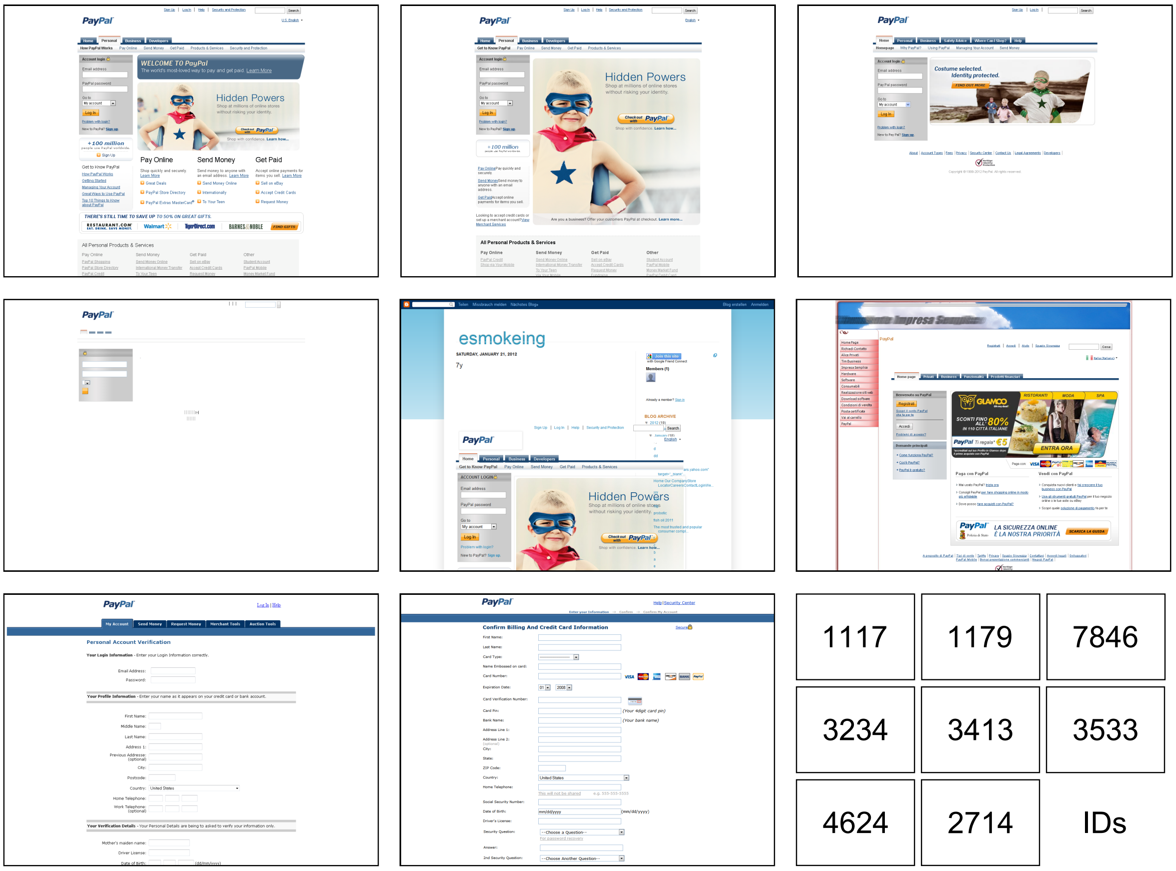


Figure 5.4: Most attacks try to closely mimic the appearance of the parent website but some also different to the original, either due technical limitations of the phisher or because of content variations.

PayPal are used for the text contents (without displaying the PayPal logo anywhere). This could be either to subconsciously place another brand cue for the user visiting the website or perhaps the phisher only reused the styling information of another PayPal phishing attack he launched earlier. We also observed some cases that tried to gather credit card data without phishing one specific credit card operator. ID 1458 shows an example for a website using the MasterCard and Visa logo at the same time.

5.1.6 Application of The Test Set

Having a large test set such as the one presented in this chapter should make it possible to perform comparable tests among different detectors that make use of URL, HTML or graphical properties. Most of the detectors that have been presented in the related work chapter (see section 3.5) could hence be tested with this test set making performance results comparable more easily.

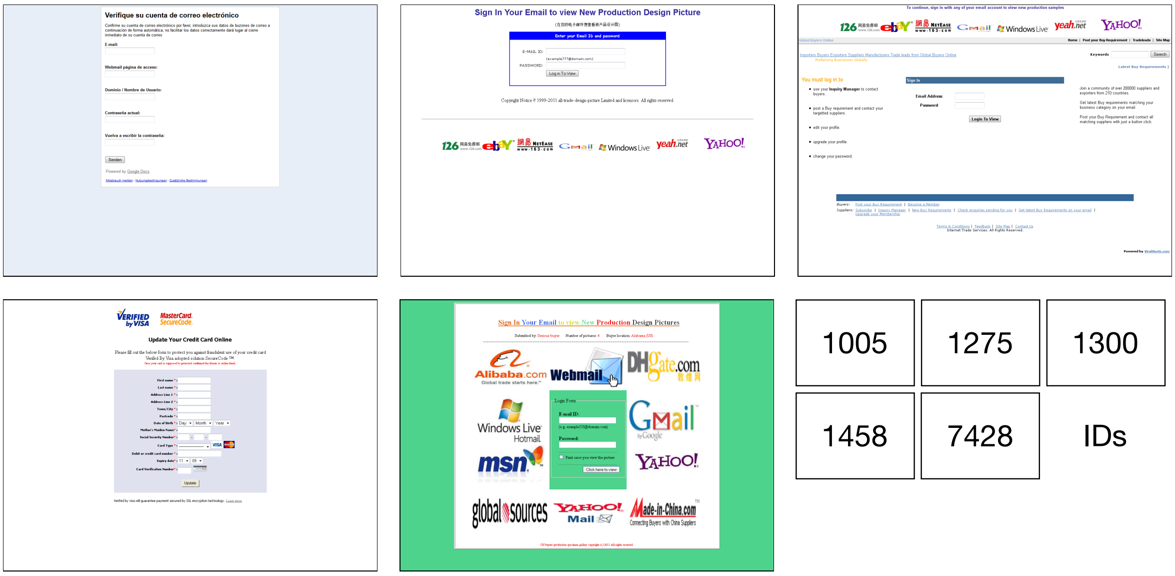

Verifique su cuenta de correo electrónico
Sign In Your Email to view New Production Design Picture
Enter your Email ID and password
Log in To View
To continue, sign in with any of your email account to view new production samples
Sign In
Login To View
Update Your Credit Card Online
Update
Sign In Your Email to view New Production Design Pictures
Login Form
Click here to view
1005
1275
1300
1458
7428
IDs

Figure 5.5: Different screenshots for websites of the state “Phishonly” (2) that target either no specific or a multitude of different brands.

The contained phishing and non-phishing websites make it possible to not only test the detection rate of phishing attacks but also look out for false positives and true negatives delivered by a detector. Using the linkage with the corresponding brands researchers may also test detectors that predict the similarity with an original counterpart of a phishing website.

To facilitate this we made the test set publicly available on our website for other researchers to download. A MySQL dump of the different database tables can be downloaded as a compressed archive to import it into own databases. The website as well as the archive also contain explanations on the different database tables that are contained within the test set.

Although a standardized test set allows for better comparison of different detectors it also has its limitations. As phishers and their methods evolve, a phishing test set should be as up-to-date as possible to make sure that the detectors are not tested against already outdated attack types. In this case comparability conflicts with up-to-date-ness and researchers should make up their mind whether its better for them to have a comparable basis to other research which they could easily reuse or an accurate test set that most probably will need to be collected by themselves. A possible solution to suit both dimensions best would perhaps be to offer a standardized way of collection phishing attacks at every moment (e.g. offering a phishing collection software). Like this, researchers could collect up-to-date data that has the same properties as data sets of older research. For comparability they could then do their tests with the test set of previous research or if applicable retest the work of other researchers with their own new test set. Perhaps the parameters that were used for the creation of this test set and the database table structure could serve as such a standardization with some slight extensions.

Some detectors from related work used additional properties that were not captured within our test set. Domain-based information (like the time of registration) are not part of our test set, as well as only the main HTML corpus of the first page was stored in our database. If information of additional images, style resources or other websites linked from the main website were needed for the detection process they would need to be added to the test set as well.

5.1.7 Research Results

Although the creation of this test set can be seen more or less as a basis for further research, its definition and the collection already brought up some research findings which contribute to the research questions covered within this thesis.

DD *What is Phishing Detection?*[8]

When building up a phishing test set the dimensions and the definition of phishing detection play an important role. In terms of this chapter phishing detection or a phishing detector could be defined as a piece of software that is able to classify website input correctly into phishing and non-phishing attacks.

In other words, in each test of a phishing detector using a test set, the test set represents the definition of phishing and how the world of phishing looks like towards the detector. Looking at properties of our and other related work one can see phishing detection as a function mapping different technical details about websites to a binary classification problem of whether a website is phishing or not. In the test set presented, some of these input properties have been gathered (e.g. URL, HTML content, screenshots) and some have been omitted (e.g. domain information, linked content).

$$f\,(\text{attributes(website)}, \text{context info}) = \begin{cases} 1 & \text{if is phishing} \\ 0 & \text{if is NOT phishing} \end{cases}$$

$$\text{attributes(website)} = \begin{pmatrix} \text{URL} \\ \text{HTML content} \\ \text{screenshots} \\ \text{domain info} \\ \text{linked content} \\ \dots \end{pmatrix}$$

Depending on the type of detector the returned result may be more than just a binary return value. Additional output information like a probability score instead of a solely binary classification can make it easier to communicate the detection results to the user or use them for further processing.

[8] For each project subchapter the results concerning the different research questions can be recognized by the respective shortcut icons from the research question overview.

Another perspective on what phishing detection is, can be gathered through the analysis of a phishing test set. What kind of data is phished? What types of businesses are attacked? Using such observations does not only give an overview over what phishing is but also on what phishing detection needs to be. The details provided in this chapter should already provide an overview over the current state-of-the-art of the attacks.

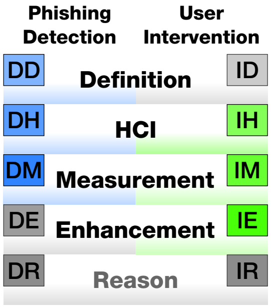
DM *How can Detectors Be Evaluated?*

To evaluate detectors test inputs are essential to find out whether the detector works as intended. Feeding those test items into the detector and comparing them to the desired results yields the benchmark data that is used for rating the detector. Does the detector produce a lot true positives and true negatives or are there a lot of false positives and false negatives? Again the test set is vital to make sure that all general and corner cases of possible phishing attacks are covered.

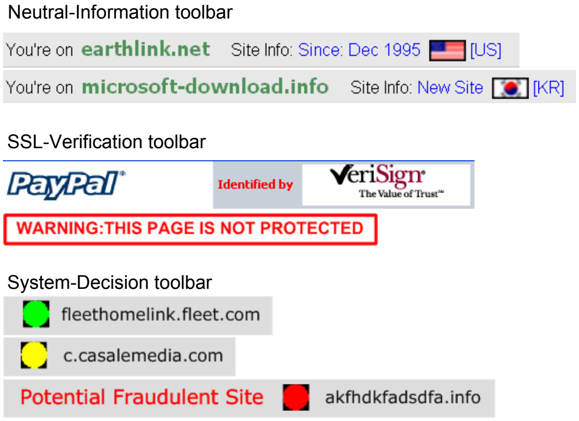
As mentioned in section 5.1.6 such a test set could either be standardized for better comparability of the detection results of different research approaches or should be up-to-date to cover the recent phishing landscape. For the best evaluation possible both should hold true. A test set as it has been presented in this chapter can be seen as a first step towards such a goal. It does not only provide a public testing basis that can be reused by other researchers but also provides standardized structure of possible input variables to detectors. Reusing this model would make it possible to use different test sets of different time spans as inputs to different detectors to provide the best comparability between detectors.

5.2 SecurityGuard Website Status Rollup

This chapter is based on the work that was part of the diploma thesis "Enhancing Web Browser Privacy and Security Awareness" by the student Dominik Andreansky [13].



The goal of the project that is described in this subchapter was to create a web browser status indicator that would be similar to other status indicators (e.g. security toolbars) that have been previously proposed and deployed but that mitigates the downsides of such approaches that have been shown in related work so far (section 3): first of all most users usually overlook non-blocking status indicators simply because security is not the primary goal of their actions. Within the developed extension we try to counteract this using a large customizable browser sidebar that provides continuous feedback on many different parameters and, for maximum visibility, by coloring the whole browser depending on the overall security. A second problematic property of such status indicators is that users are often unable to understand the information that is conveyed by the indicator and hence are unable to take the correct decision. Here it can help to offer security content relevant to the user that is presented in a way it is easily understood.



Neutral-Information toolbar

You're on earthlink.net Site Info: Since: Dec 1995 [US]

You're on microsoft-download.info Site Info: New Site [KR]

SSL-Verification toolbar

PayPal Identified by VeriSign The Value of Trust

WARNING:THIS PAGE IS NOT PROTECTED

System-Decision toolbar

🟩 fleethomelink.fleet.com

🟨 c.casalemedia.com

Potential Fraudulent Site 🟥 akfhdkfadsdfa.info

Figure 5.6: Screenshot of the passive warning used by Egelman et al. [78].

Figure 5.7: Three different types of toolbars used in the study by Wu et al. [324].

We developed a Firefox extension (called "SecurityGuard") that displays diverse security related information on the user's visit to websites within several information areas in a browser sidebar. Each information block contributed to an overall score that finally set a general security recommendation. The extension was designed with a user centered development approach using an online survey and a paper prototyping phase. We finally evaluated the plugin in field study by recording usage patterns from 24 plugin users.

5.2.1 Yet Another Status Toolbar?

When looking at the related work in terms of computer security warnings in the browser (see section 3.3.2) researchers found that non-blocking indicators and especially security toolbars don't seem to work. Egelman et al. [78] found that in case of passive browser warnings the users did not have more success in evading security problems than a control group (without any warning). Only 30% of the participants read the warning at all, because some accidentally dismissed it during typing on the keyboard. The passive warning they tested (see figure 5.6) was a non-blocking popup-dialog. The only permanent visible component of this indicator was a small text on the right side of the URL bar. Wu et al. [324] tested several browser toolbars with similarly small indicators. They distinguished between a "neutral information toolbar" (only showing more information about the domain), a "SSL verification toolbar" (showing additional information about the SSL protection) and a "system decision toolbar" (using a traffic light metaphor to propose how secure the website is) (see figure 5.7 for the three toolbars). In their tests the highest number of participants fell for attacks when using the neutral information toolbar (45%) whilst 33% still fell for attacks using the system decision toolbar. The explanation given by the authors is that the participants also failed to look at the toolbars in many cases and solely relied on the content area of the browser for making their judgments.

Looking at all these facts, one has to ask why one should bother creating another non-blocking information tool for user intervention. We saw four reasons why more research in this area might be promising: 1) looking at the previously described research one can easily see that the systems tested where more or less at the minimum of the range of possibilities that could be tested. Regarding the passive warning of the Internet Explorer it is clear that such small indicators will be overlooked by the user (as previous research already proved that the lock icon was never looked at). In case of the toolbars the design of these toolbars was very rudimentary and perhaps did not stand out enough of the rest of the browser's user interface. 2) Besides the problems that research sees with toolbars they are still used as a widespread instrument of companies for customer information and customer security. Rand Fishkin [90] compiled a list of 14 different popular browser toolbars. 3) Blocking security messages have the advantage that they cannot be overlooked but that comes with another problem: if the user actively wants to disregard them (e.g. because they are erroneous) this can get very time consuming. The untrusted SSL certificate warnings of Firefox are an interesting example as they often need to be circumvented to reach self-signed SSL secured websites. 4) Another advantage of a permanently visible indicator is that it is not only able to provide feedback in case of errors, instead it can also be used to constantly reinforce the user positively on standard websites.

For our "toolbar" presented in this subchapter we chose to do some radical changes compared to the toolbars tested so far. We tried not only to enlarge the area of toolbar feedback within the browser to a maximum but also raised the amount of reported information. In the end, we included website identity information, blacklist information, server location and domain information, password and visit information, and gave an overall summary and recommendation that led to complete browser design changes (see figure 5.8 for example screenshots). Before explaining the different components in more detail we will start off with the user centered design process of the toolbar that we took.

5.2.2 Designing the Extension

Related work and other prior research already showed possible directions where another web browser extension should go or rather where it should not go. This already gave us a broad idea of what a new toolbar-like extension could look like. To gather more details about user expectations towards such a tool we first conducted an online survey. Afterwards we wanted to gain user interface design insights doing some paper prototyping sessions, before we arrived at the final design of the "SecurityGuard" browser extension.

Online Survey

To gather insights about user expectations towards such a toolbar we designed a 32 question online survey. Besides some demographic questions, we asked people to describe and explain existing warning dialogs; about their knowledge on SSL certificates and phishing

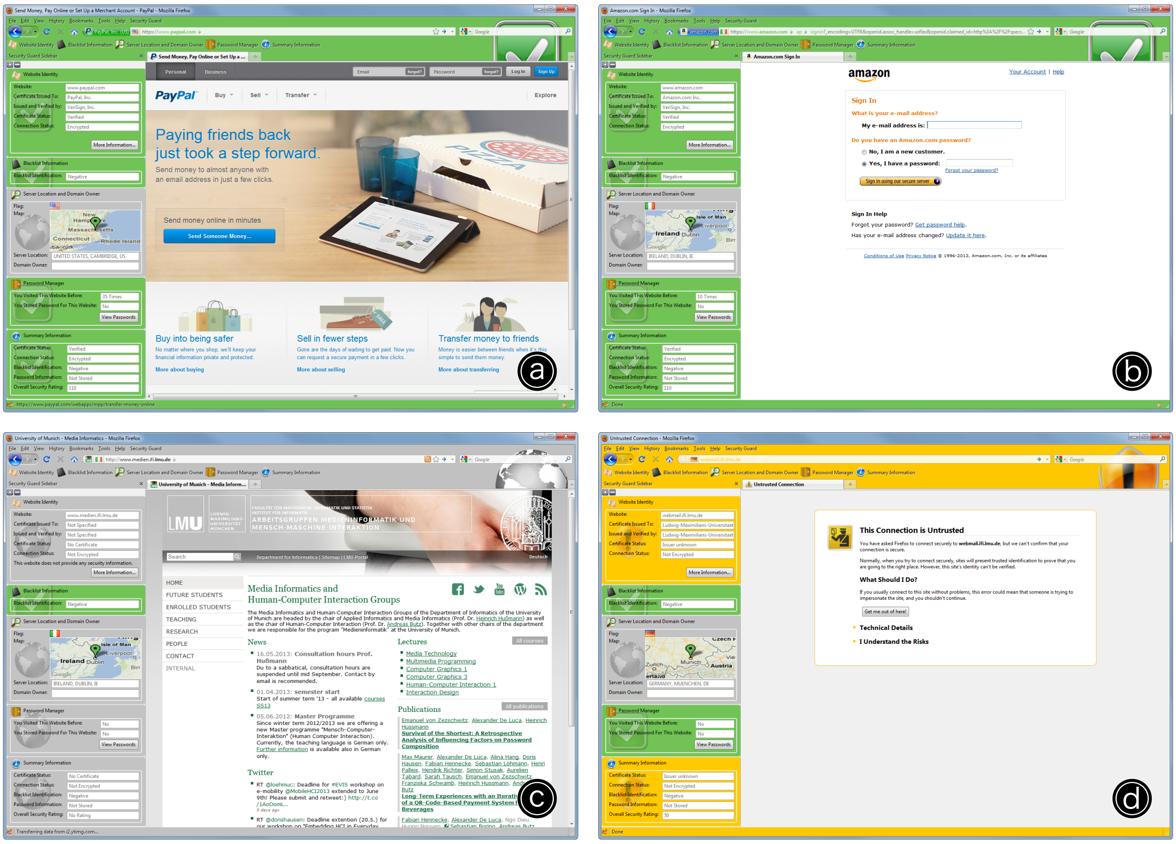


Figure 5.8: Four screenshots of the look and feel of different states of the SecurityGuard extension. a) Revisiting an extended validation protected SSL website (paypal.com); b) Revisiting a standard SSL secured website (amazon.com sign-in); c) visiting an unsecured previously unvisited website (lab website); d) visiting a self-signed SSL protected website for the first time.

and about their feelings about certain design criteria for security dialogs (colors, icons). Finally we asked several questions about their behavior when seeing such warnings and about preferred wording of security related messages.

The questionnaire was promoted online and received 49 complete responses within 14 days. In average the participants' age ranged from 17 to 58 years (31 years in average). 24 participants were female. 22% stated to have "excellent" or "good" online security knowledge. Only a small fraction of people were able to correctly explain the contents of the presented warning examples. Some even thought the screenshots we had included were fake. 47% stated to know what SSL certificates are, whilst only 31% were actually able to explain them correctly. The term phishing was correctly explained by all 73% that stated to know what the term means. 12% of our participants actually used an anti-phishing tool.

Looking at colors and iconography that was preferred by the users the standard color expectations of a western culture were met. Green was selected most (43%) as a metaphor for security and red (43%) as one for insecurity (see figure 5.9). The participants were allowed

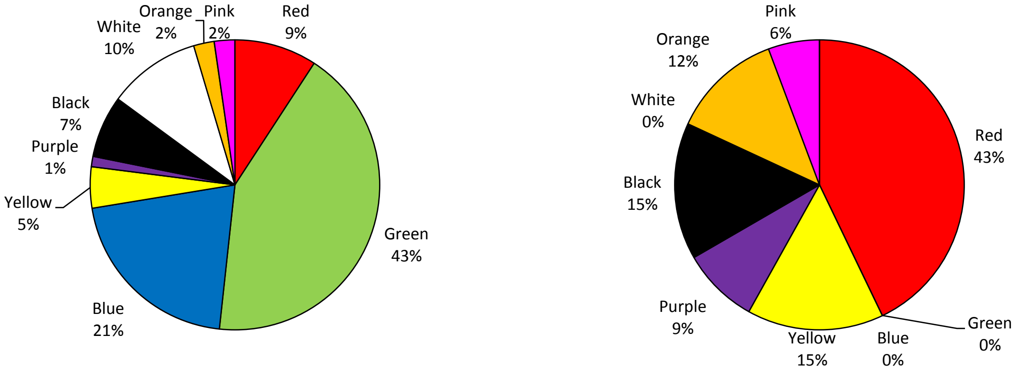


Figure 5.9: Left: Ratings for colors perceived as "secure" by the participants. **Right:** Ratings for colors perceived as "insecure". It was possible to select multiple colors.

to select more than one color. In terms of iconography the current icons of the existing Firefox and Internet Explorer web browser were rated superior to the the classic lock icon. A smiley icon performed worst.

Only 37% of the survey respondents stated they would not read browser warning messages because they are too long or technically phrased. In case a warning would ask them to leave a website 67% of the participants stated they would do so.

As we planned to use a rather large amount of screen real estate for our extension we also asked people how much screen space they would be willingly give up for a security plugin. More than half of the people (53%) would only allow for up to 10% of the screen space to be occupied by security indicators. 27% would sacrifice 15% of the screen space and 14% even 20% of their screen space.

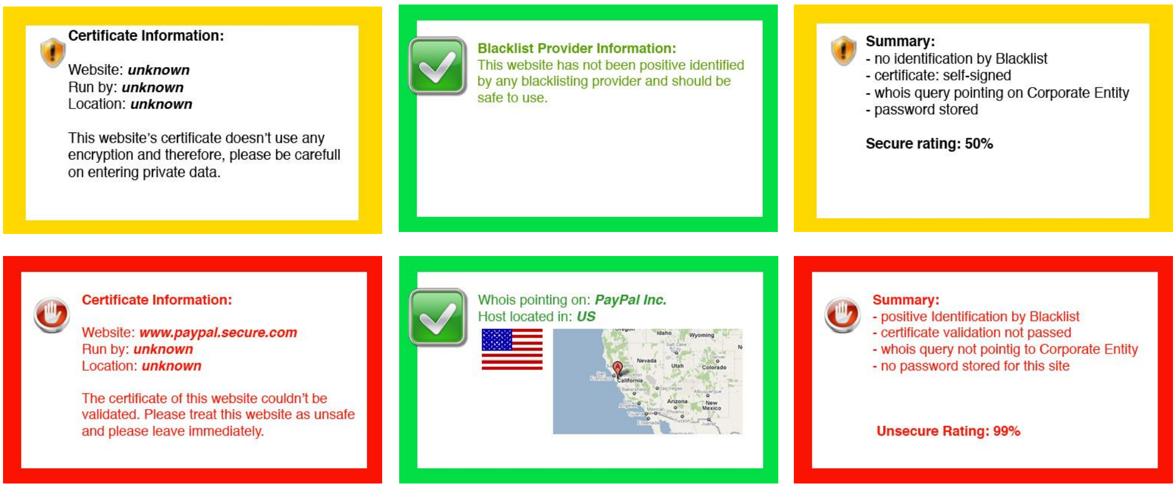
Paper Prototyping

With the results of this first survey study we created a first draft of our plugin concept and decided that it should be implemented as a browser sidebar that is divided into different information modules.

To get a more detailed feedback for our concept we decided to build some paper prototypes to collect first user feedback. We built a few simple mockups for a range of possible modules that could be used in the later plugin (see figure 5.10) and used three different states for each module (green for secure, red for insecure and yellow for suspicious or missing information).

We presented them together with mockup browser chromes to nine participants in single interview sessions and explained the concept of the plugin and of the different modules to them. Participants were asked to play around with the concept and arrange the modules as they liked. We also provided different browser contents (warnings and real content) as puzzle pieces to build a whole browser mockup.

The feedback from the participants was diverse. The participants preferred different levels of information details. One participant was only interested in a security summary and wanted



Certificate Information:

Website: *unknown*
Run by: *unknown*
Location: *unknown*

This website's certificate doesn't use any encryption and therefore, please be carefull on entering private data.

Blacklist Provider Information:
This website has not been positive identified by any blacklisting provider and should be safe to use.

Summary:
- no identification by Blacklist
- certificate: self-signed
- whois query pointing on Corporate Entity
- password stored

Secure rating: 50%

Certificate Information:

Website: *www.paypal.secure.com*
Run by: *unknown*
Location: *unknown*

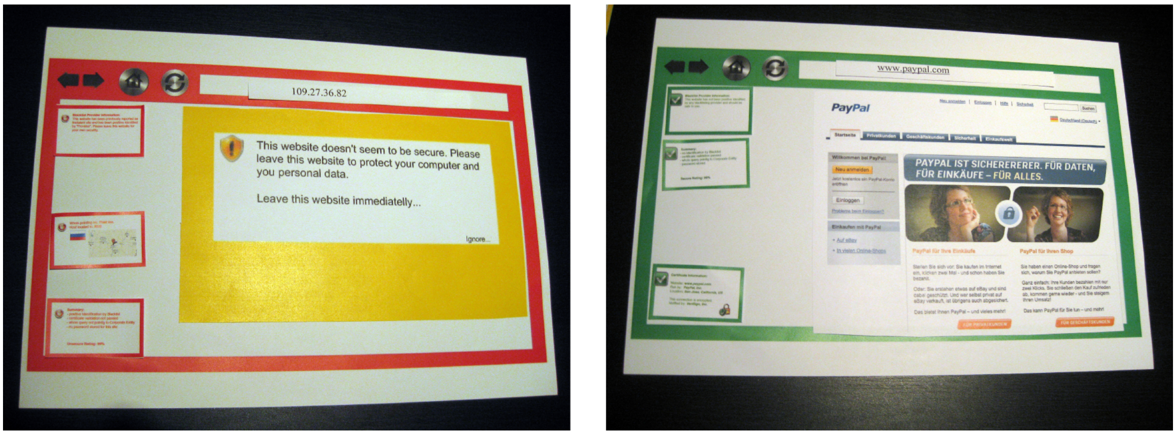
The certificate of this website couldn't be validated. Please treat this website as unsafe and please leave immediately.

Whois pointing on: *PayPal Inc.*
Host located in: *US*

Summary:
- positive Identification by Blacklist
- certificate validation not passed
- whois query not pointig to Corporate Entity
- no password stored for this site

Unsecure Rating: 99%

Figure 5.10: Example modules used for the paper protoyping study.



109.27.36.82
This website doesn't seem to be secure. Please leave this website to protect your computer and you personal data.
Leave this website immediatelly...
Ignore...
www.paypal.com
PayPal
PAYPAL IST SICHERERERER. FÜR DATEN, FÜR EINKÄUFE – FÜR ALLES.

Figure 5.11: Two example pictures of the paper prototype browsers that had been compiled by the participants. One using the scenario of paypal.com as a website and one using the scenario of a simple IP-address.

to access detailed data on request. Another participant was interested to see the blacklist and SSL certificate status. During the interview we asked our users to build browser configurations for three different URLs (the www.paypal.com URL, an IP address and a arbitrary university subdomain “something.lmu.de”). The participants made up their own scenario of what the URL could mean and configured the paper prototype accordingly. The scenario also influenced the choice of modules. Figure 5.10 shows two example paper prototypes that have been compiled by participants of the study.

We also discussed some general design decisions with the participants. Participants had no clear preference of whether such a sidebar should be docked to the right or the left of the screen. Some wanted the sidebar to be always present, whereas another participant wanted it to fade in with each new page load and to fade out as soon as interaction within the content area was started. Another participant wanted to have different module configurations depending on the summarized state of all modules.

Final Extension Design

Accumulating all findings from related work, our online survey and the paper prototyping interviews we ended up with a final design concept for our SecurityGuard extension consisting of four general browser elements:

- **SecurityGuard Sidebar:** The major component of our extension is the browser sidebar containing the different information modules. The whole sidebar and each module could be collapsed and reopened to support each configuration the user wanted. The standard configuration was to show all possible modules to the user. Depending on the module some of the following module states are possible: a green state symbolizing that the partial result of the security assessment of this module is good; a yellow state that indicates some problems or missing information; a red state indicating security problems within that module and a gray state indicating that this module does not have a security judgment. The sidebar consisted of the following five modules:
  - **Website Identity Information:** This module shows the current state of the SSL certificate, consisting of the website the certificate was issued for, the name of the company and the issuer, the certificate status (whether it was verified) and finally a connection status indicating whether the connection is encrypted. Using a button with the caption “more information” the user can open up the browser’s security information dialog for even more information.
  - **Blacklist Information:** This module contains only a single information indicator telling the user whether the website appears on a blacklist index.
  - **Server Location and Domain Information:** This module collects several pieces of information of the server and the domain the user is connected to. First of all the geo-location of the server’s IP address is retrieved and a flag, a map and a textual representation of the location of the server is given. Besides this the person or company that has registered the domain name is displayed.
  - **Password and Visit Information:** This module displays the number of times the user has visited the website and whether the user has stored a password in the browser’s password manager for this domain.
  - **Information Summary:** Finally the information summary module gives a short summary over the most important states of the other modules and displays an overall security rating that is calculated by the extension.
- **SecurityGuard Toolbar:** The security guard toolbar consists of five different buttons each one belonging to one of the modules in the sidebar. It can be used to quickly switch modules in the sidebar on and off. We introduced this possibility as some users

wanted to modify the information display when necessary. It is not necessary to have this toolbar enabled to make use of the SecurityGuard extension.

- **SecurityGuard Location Bar:** The SecurityGuard extension also enhances the display of the location bar. Next to the browsers own SSL status indicator a flag showing the country where the domain's server is situated is introduced. This indicator is identical to the one in the server location module. In addition to that the style of the URL display is also different to the usual appearance in web browsers. The URL is formatted in a color, matching the global security rating of the website (as identified by our plugin). The basedomain-part of the URL is formatted differently to the subdomains and the scheme of the URL (e.g. "http" or "‘https"). Each folder of the path of the URL is separated using small icons (as known from breadcrumb navigation[9] in websites). All this highlighting and separation should make the different parts of the URL easier to understand for the user to make it easier to recognize URLs that impersonate another domain somewhere in the subdomains or the path for example.

- **SecurityGuard Personas:** Finally the browser's persona is also changed to resemble the color of the security decision of the plugin. This helps to change the complete appearance of the browser without using more screen real estate. Besides using this idea within this extension for the first time, we conducted a follow-up project (see subchapter 5.6) to examine the possibilities of this kind of user intervention more closely.

Figure 5.12 shows a labeled example screenshot of the final browser extension. A figure showing different states of the module for different example websites can be found in figure 5.8.

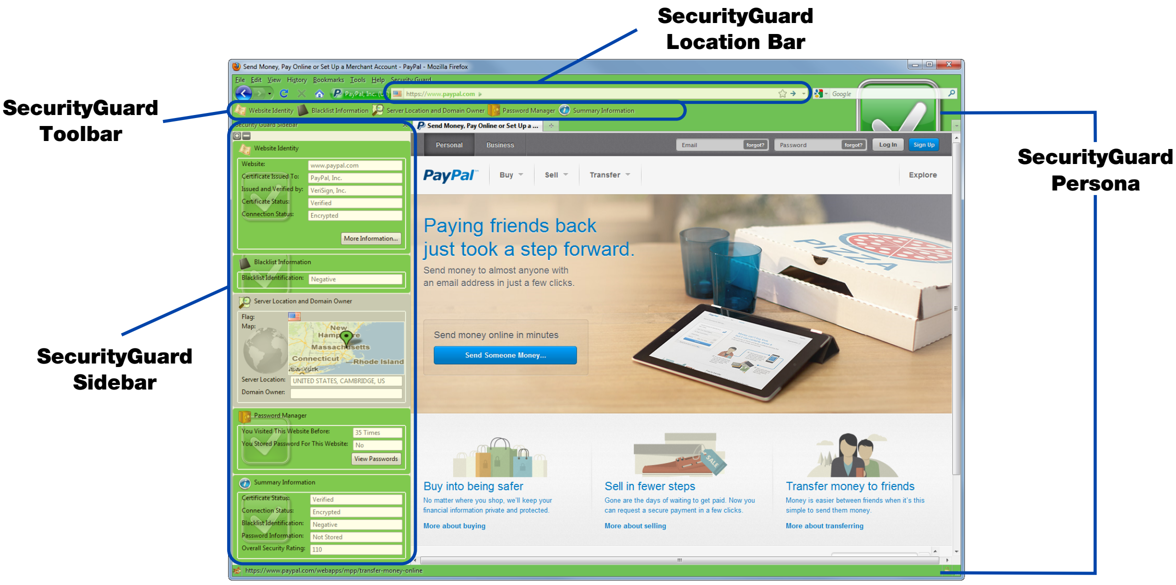
5.2.3 Implementation

The SecurityGuard extension has been implemented as a Firefox plugin using the standard hooks and methods of the Firefox API[10]. When the browser is started the plugin registers two hooks to browser events to start the validation of the current page. One is fired with every new page load that happens, the other one is fired whenever the active tab of the browser window is changed (because the userinterface components of the browser need to be updated then).

Whenever the security information needs to be refreshed, different functions are called that update the information of each module separately. As some modules have to reload information from outside sources the overall security information is updated after all information is fetched.

[9] Breadcrumb navigation uses the idea of Hansel and Gretel of having a trail of breadcrumbs to find their way back in websites [160]. A path of a URL can also be seen as a trace of sublevels back to the top folder.

[10] `https://developer.mozilla.org/en-US/`



Website Identity

Website: www.paypal.com
Certificate Issued To: PayPal, Inc.
Issued and Verified by: VeriSign, Inc.
Certificate Status: Verified
Connection Status: Encrypted

Blacklist Information

Blacklist Identification: Negative

Server Location and Domain Owner

Flag:
Map:
Server Location: UNITED STATES, CAMBRIDGE, US
Domain Owner:

Password Manager

You Visited This Website Before: 35 Times
You Stored Password For This Website: No

Summary Information

Certificate Status: Verified
Connection Status: Encrypted
Blacklist Identification: Negative
Password Information: Not Stored
Overall Security Rating: 110

Paying friends back just took a step forward.

Send money to almost anyone with an email address in just a few clicks.

Send money online in minutes

Buy into being safer

No matter where you shop, we'll keep your financial information private and protected.

Sell in fewer steps

Gone are the days of waiting to get paid. Now you can request a secure payment in a few clicks.

Transfer money to friends

Money is easier between friends when it's this simple to send them money.

https://www.paypal.com/webapps/mpp/transfer-money-online

Figure 5.12: The different parts of the SecurityGuard browser extension.

The SSL status information can be read directly from browser objects identifying the different SSL status and information about the SSL certificate. The blacklist information is received using an API call to the Google Safebrowsing API. The server location is received from another free API that allows to resolve geographical details for IP addresses[11]. The flags are displayed using flag graphics we bundled with the extension whereas the map representation was loaded dynamically as a query to Google Maps[12] returning a static image. The domain owner is retrieved from another API that can be used to receive WHOIS-information[13]. The number of visits to the current website and the number of stored passwords again could be extracted from standard calls to the Firefox API.

5.2.4 User Study

To evaluate our extension we chose to conduct a small field study. We added some logging functionality to the plugin and made the plugin publicly available on a website together with installation instructions in German and English. The main purpose of our study was to find out how the users would make use of the different modules and the configuration options they had at hand. The initial online survey suggests that people are not willing to have a big portion of their screen occupied by a security related information plugin. We wanted to check whether this would be true for the our plugin in the field.

[11] http://www.ipinfodb.com

[12] http://maps.google.com

[13] http://www.kahtava.com

Methodology

We asked the participants of our field study to download and install the plugin and use it for a period of two weeks. The log data was not submitted to our server but instead we stored it locally on the participants computers and asked them to send the information back to us after the study period. We also did a small exit interview with them, followed by an additional online survey.

We recruited 24 participants excluding people that had previously participated in our paper prototyping study to prevent biasing. During the collection phase we were mostly interested in the kind of interactions the participants did with the browser extension. Hence every interaction with the plugin was stored within the log files noting the respective element, the action, a timestamp, and the user interface component from where the action was triggered. We did not store any privacy related data like website URLs, but recorded more general data like start and end of a browsing session and each time a sidebar module was opened or collapsed.

Results

The average age of the 24 participants of our field study was 31 years with 75% of the participants being male. The participants were recruited using posts on a social network platform.

In total we analyzed 680 different webbrowsing sessions (in average 28 per participant in 24 days). Looking at the log files we saw that 21 of our 24 participants (88%) collapsed or hid the sidebar at some point during the field study whilst only three had it open for the complete duration of the field study. Of the 21 people that collapsed the sidebar only six (25%) kept it closed at all times. The 15 remaining participants opened some single modules or the whole sidebar whenever desired. In total, we monitored 310 openings and closings of either single modules or the complete sidebar. In contrast to what we expected, the toolbar buttons were hardly ever used, instead the users collapsed the sidebar entries using the section headers within the sidebar.

Looking at the numbers how often the sidebar or single modules have been opened or collapsed, the whole sidebar was most often opened (89 times by 18 participants). Looking at the single modules the number of open interactions are only close behind. The map module was opened 69 times by 12 participants, the blacklist module was opened 65 times by 13 participants, the website identity module 52 times by 15 participants and the password module 46 times by 10 participants. Most interestingly we found only 28 open interactions for our summary module. This could be because of the fact that the overall security assessment was also shown using the browser persona and other indicators but it could also mean that the module was less often collapsed.

We didn't expect that the map module was requested this often. In our paper prototyping study it was hardly used and hence we also excluded the contained data from the overall security value calculation.

Asking the participants in the interviews when and how they used the SecurityGuard extension, 79% stated that they had a look at the modules whenever they did financial transactions online, or when the persona (67%) or one of the modules (50%) turned to another color than green or gray. 13% were also sometimes just curious about the contents of the modules.

Besides demographic data the final questionnaire contained mostly Likert scale questions belonging to three different categories. We used five point Likert scales ranging from “1 - strongly disagree” to “5 - strongly agree”. We had 22 different questions of which we will only report on the most important ones of all three categories (see figure 5.13).

The first category of questions dealt with the experienced usability of SecurityGuard. All but one participant found that the extension was easy to use and easy to learn and nearly all of our participants were overall satisfied with the extension. In terms of understanding the extension again high ratings where achieved whilst the overall layout was rated worst with one person disagreeing and one person being neutral about the fact that the layout was easy to comprehend. We think that the wealth of different information options that is offered by the plugin might be a reason for this. The information of the single modules was well understood by all participants. Asking whether participants understood the color coding they reported to understand all coding equally well. We thought that the fact that different colored modules then are summed up to an overall recommendation might possibly confuse our users. The last set of questions asked the people about their experienced security when using the plugin. Again the participants thought that security helps them to find online risks and that it helps them understand security concepts better. Looking at the median rating values, so far all questions had a median rating of 5. The question about the understanding of security concepts was the first one that had, despite a high average, a median value of 4. We think this is reasonable as although our extension gives insights to a lot of different security related parameters we did not focus on the specific explanation of the different security concepts. As a last question we asked the participants whether they would continue using our extension after the field trial. Seventeen agreed with that while four people disagreed and three other people answered neutrally. These numbers fit well with the six participants that had not done any major interactions with the sidebar during the study phase.

5.2.5 Discussion and Limitations

Within this project we used a user centered development approach – including online surveys and paper prototyping – to build a browser extension providing security related details to the user. Looking at our results it seems that at least for some of our test users and modules we were able to get the users to seek advice in the information provided by our tool. This is in slight contrast to what related work reports, that such non-blocking warning interfaces fail to catch the users’ attention. In some cases the users even looked at the results out of pure curiosity without any security related questions in mind. Such an approach to security information display could make the topic of security much more attractive than it is today. If

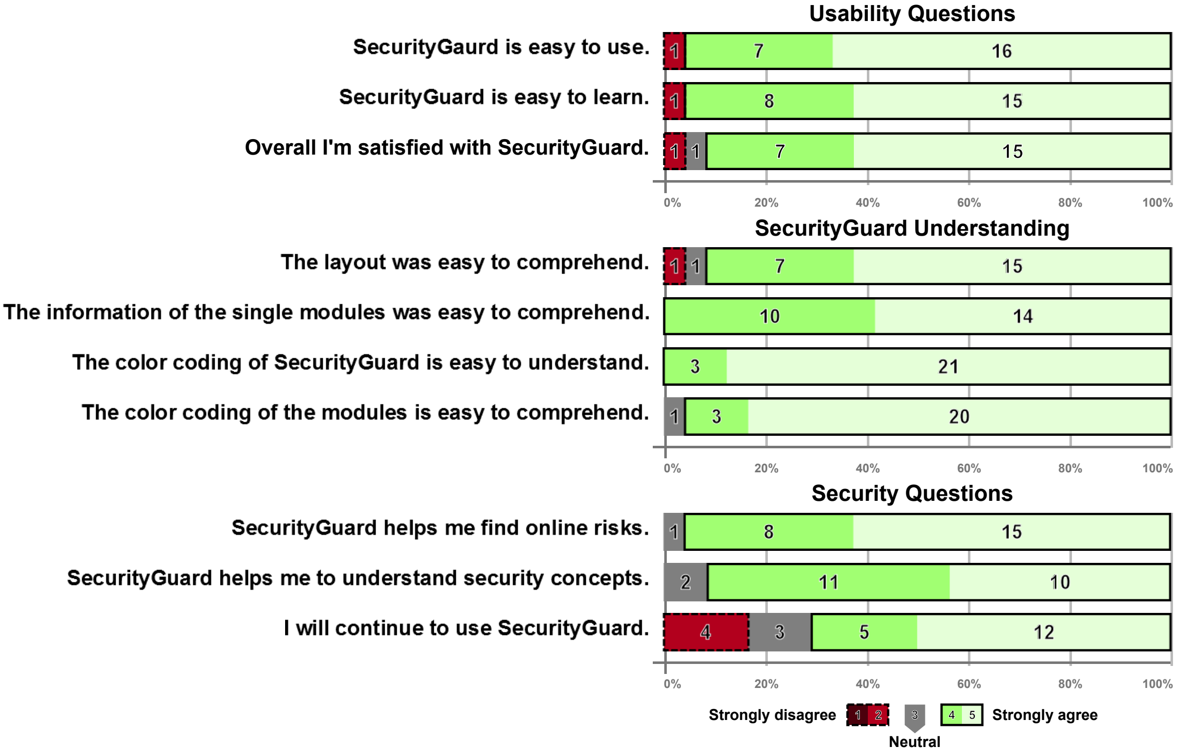


Figure 5.13: The answers of participants to the Likert questions for the SecurityGuard field study. We asked questions about the usability and understanding of SecurityGuard, as well as security-related questions.

users familiarize themselves with security related topics in other contexts it could help them to better understand security problems.

Looking at our development process the results from the pre-studies in many cases predicted the final outcomes in our field study. However, the insights where pre-study and field study did not match seemed to be the most interesting. Although the server location and information did not matter much to the participants of the pre-study, the field study candidates looked a lot at this data. In practice malicious websites can be hosted on any server in any country, but the sole fact of people starting to reason about the technical properties of their actions is a step in the right direction. The form of presentation also might be a major point that made this security module more interesting than the others. A country-flag and a location on a map is something a user can easily identify with. Do I know that place? What are my feelings about this area? Does it make any sense that this company hosts their data there? These are all questions that could arise when looking at such a representation.

Besides the positive qualitative and quantitative resonance that we had when analyzing our log data, still 25% of the participants nearly completely neglected the sidebar as a core component of our plugin. Haven't they been interested in security at all or was the overall rating displayed by the persona already enough for their judgment? Looking at the qualitative answers most of them seemed to be happy with the extension. In future studies it would be important to take extra care of those participants to find out exactly what their reasons were.

Collecting data on the individual module usage of our participants on the one hand gave a very good insight on how our plugin was used. On the other hand, we did not evaluate in how far our concept really would protect users that visit malicious websites. In case of this project we wanted to collect knowledge about which types of information are interesting towards the user and how she can be reached. We think that a field study evaluation was suited very well to achieve this goal. A measurement of how often a user falls for malicious websites usually needs to be done within a lab study. In an upcoming project (see subchapter 5.5) we tried to solve these problems by performing an extra lab study to find out about how participants would perform when being exposed to malicious websites and collected anonymous website visit data throughout our field study. This made it possible to get more insights about the actual browsing behavior of the participants without affecting their privacy.

Another limitation of this project was that we did not do any evaluation of our scoring system. We hence have no results about the correctness of the recommendations given by our plugin. Without those quantitative values on detection a comparison to other existing approaches is close to impossible. Looking at the interview comments and the quantitative and qualitative results of our study it at least seemed like we were able to influence the security perception of the users using our plugin. The sole action of opening a module to check its contents already shows that we were able to raise the interest towards security and hence influence the security perception in some way. As the persona concept has been very well accepted within this project we will focus on this specific aspect more in the project in subchapter 5.6 to see whether we can influence the users' security perception using such methods.

5.2.6 Research Results

This project focused mainly on the user centered development of a new kind of user intervention system. Within this project we didn't focus on the quality of the detector but were instead interested to find out which kind of information in which kind of representation would be interesting for a user. In the course of this project we kept this detector mechanism to the minimum necessary to have enough data for our desired outputs. This shows that working on new ways of user intervention nearly always introduces some findings related to the detection of attacks.

DD *What is Phishing Detection?*

The usual way of the definition of phishing detection as it is presented within this thesis includes a measurement of how well a given detector performs in detecting phishing attacks and other websites. In case of this project we had a lot of different modules that each produced a recommendation that was then finally summarized by an overall module. This raises the question what level of transparency should be applied to such detection results. Is it better to hide everything but the final detection result from the user and try to automate the decision process or does a variety of possible information eventually have some advantages? Each participant in the evaluation steps throughout this study had different ideas of what information he or she would want to see. This means that in case a user is able to make use of different detection stages he should perhaps be incorporated into the detection process.

DH *How can HCI be Used to Build Detectors?*

It seems that for a proper protection of the user against malicious websites a lot of individual factors have to come together. It is not only about how well the technical aspects lead to a good phishing detection, it is also about whether the user can understand the problem and data that is finally reported and whether she is interested in it. In other words a detector development process already needs to have knowledge about the users of the final system. In case of this project the user centered design approach first told us what the users would like to know to judge the security of a website and how they would like to see it represented. We then built a minimum detector to fulfill what we identified that would be necessary for the desired user intervention mechanism. This way of reversing the development process of a phishing detector may help to come up with new ideas that will most likely be better understood by the final user.

DM *How can Detectors Be Evaluated?*

We did not perform an actual measurement of the quality of our simple detector by checking its results against malicious and non-malicious websites. Although such an evaluation should usually be performed to find out details about the technical benchmarks of the detector the evaluation of the user intervention system can also yield findings that serve as a basic detector benchmark. Detectors with a low detection quality will always lead to inaccurate user intervention that will sooner or later be detected by the users. Besides this, even in the final evaluation of a user intervention method interesting findings for the detector can be generated. In our case we did not use the data of the map module for our detection result as it seemed not to be important neither from the technical nor from the user intervention side. Finding that this module was the most frequently used one shows that people relied on the data and it should be incorporated by the detection process if possible.

IH *How can HCI be Used to Enhance User Intervention Mechanisms?*

Using the methodologies of software development in a user oriented way can help to design user intervention mechanisms that are much better understood and accepted. Especially in the security area where security is not the primary goal of the users actions it is important to think about what kind of intervention will reach the user best. Using a user centered design process can create user intervention mechanisms that guarantee that their contents are of some interest to the users leading to more interactions and on the long run hopefully turning the user's focus more to security.

IM *How can User Intervention be Measured?*

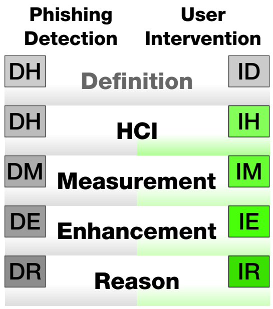
In the course of this project we did not measure the quality of our user intervention method by measuring how well people would be protected from malicious websites, instead we measured details about the plugin interaction and used a lot of qualitative values. These qualitative measurements throughout the course of the design can be very valuable to enhance specific parts of the user intervention as well as the detection. It should always be used within the development process of new user intervention methods. In case one wants to have measurements that allow better comparison of user intervention methods a qualitative measurement is needed.

IE *How to Enhance User Intervention Quality?*

Related work showed that small notifications outside of the user's primary focus are not noticed by the users and we saw within this project that occupying too much screen space is also unacceptable for most users. Yet, other users – perhaps mostly security experts – might be annoyed by blocking dialogs that appear to warn them about a potential thread. We showed that users do use and interact with non-blocking user intervention mechanisms in some way, showing that to at least some extent non-blocking warnings seem to work, too. Even when looking at the toolbar study of Wu et al. [324] one can see that the system decision toolbar performed better than the neutral information toolbar and hence had some effect. The optimal solution seems to be somewhere between both approaches (perhaps somewhat closer to the blocking dialog side). In our project in subchapter 5.5 we tried to create a semi-blocking dialog to merge the positive aspects of both approaches. Besides this our field study enabled us to reason about what properties of our user intervention mechanism were most important to the users. Although this is no final proof for the success of a concept it can be used to enhance a user intervention method before finally testing it with a broader user base in the field or measuring the protection strength within the lab.

5.3 Community-based Rating Intervention

This chapter is based on the work that was part of the bachelor thesis "Community-Based Security and Privacy Ratings for Internet Websites" by the student Simon Wicha [297]. Some parts of the project also led to a publication at 6th Symposium on Usable Privacy and Security (SOUPS) by Maurer titled "Community-Based Security and Privacy Protection During Web Browsing" [181].



Toolbar-like user intervention methods (as discussed in the previous chapter) are not the only user intervention method that is in practical use on the Internet. Services like the WOT-plugin (Web of Trust)[14] [323] offer website ratings to the users allowing each user to contribute to. Although neither the toolbars nor the community-based rating services are standard functions of today's browsers, WOT reports to have more than 80 million downloads.

Although this general concept is in practical use, the topic of community-based recommendations for security purposes has been little discussed in the research community yet. Within this project we wanted to have a look at how such a user intervention method is actually able to influence the participants' opinion and behavior. To measure this we implemented our own prototype for community-based website ratings with a security and a privacy rating. We then measured how warnings within such a plugin would affect the user's opinion on different websites.

5.3.1 The Real World Example: Web Of Trust

Before diving into the work done within our project it might be valuable to have a look at how a commercial browser intervention mechanism actually looks today. In this case we want to give a short introduction into the Web of Trust (WOT) platform to be able to compare it against our own prototype later on. While describing the current state of the WOT system – as of 2013 – here, our plugin has been developed in mid 2010 when Web of Trust already existed but was still a lot less popular.

WOT allows a community to rate websites in four different dimensions: trustworthiness, vendor reliability, privacy and child safety. Whether or not a website denotes a security problem does not have a direct mapping in the WOT scheme. It is expressed by the sum of different values instead.

After having installed the WOT plugin in the browser, it provides different status indicators to the user indicating the community-ratings for websites (see figure 5.14). For the current website that the user is visiting, a small status indicator is added to the browser next to the URL bar that opens a popup with detailed information once it is clicked. In case the average rating of a website is below a certain threshold a blocking warning message appears over the website (5.14c). For each website an online scorecard can also be accessed without having

[14] http://www.mywot.com/

the plugin installed (5.14d) and the plugin also adds the WOT indicators to links on other websites to show their status prior to an actual visit. For each rating a popularity indicator shows how many people voted for the given website to indicate how reliable the community result is.

5.3.2 Community-Based Security Research

For our research we are mostly interested in possibilities to protect users from malicious websites and hence a security judgment is most important for a plugin developed by us. Besides such a rating we also introduced a secondary privacy rating for our research.

In case of this project we wanted to use the concept of community-based ratings and a user intervention method based on this to achieve two research goals. First off, we wanted to see whether people are actually able to correctly assess those values for different websites. What is the users' definition of security and privacy and how would they rate different websites? On the reverse we were interested whether the ratings given by other people are actually able to influence the opinion of another user towards the website and whether a warning regarding security or privacy that has its foundations in the ratings of other people would be able to influence the decision making process of another user.

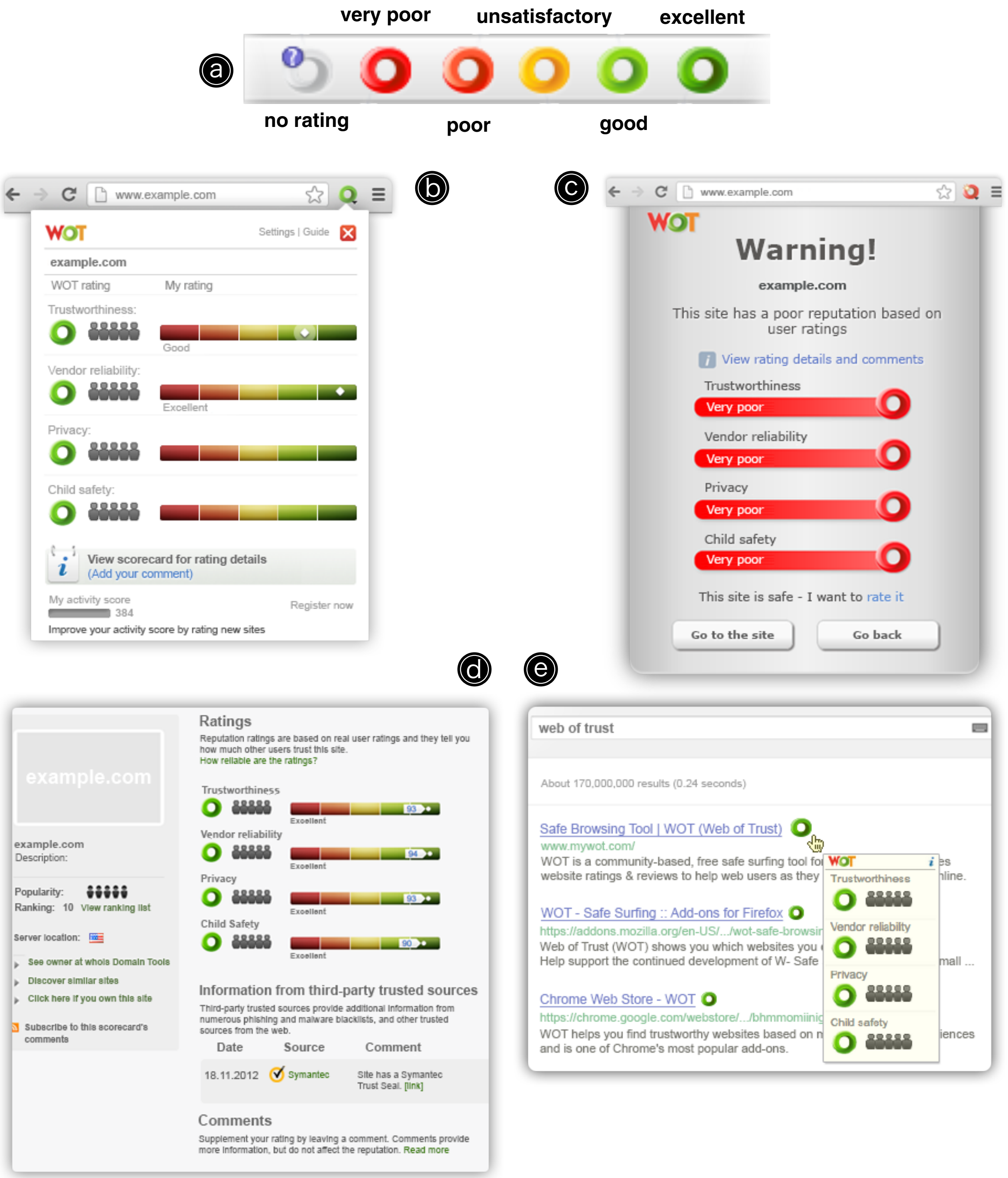
5.3.3 Building the Prototype

For building our own conceptual version of a community-based rating system we needed to develop a user interface representation in the users' browser that would be used to set and show given ratings and warnings (in case of low ratings). Besides this the plugin needed a backend server that can be used to collect and store website ratings and deliver them to other browsers as necessary.

The User Interface

Due to time restrictions we did not do any user centered design process for the user interface used within this project. After several design iterations within the team we arrived at the final designs used in our study that can bee seen in figure 5.15.

In case of average or good security and privacy ratings the only indicator that is visible to the user is a small icon together with the numeric ratings in the lower right corner of the status bar of the browser. Besides the textual ratings a small icon indicates the overall status over both ratings. Clicking on this indicator opens a popup that displays the average ratings more detailed and provides options for setting or changing the own ratings for this website. To rate the security we used five lock icons which can be clicked by the user like the well-known star ratings. To rate privacy we used an icon symbolizing a spy. Besides prompting



example.com

View scorecard for rating details
(Add your comment)

Improve your activity score by rating new sites

Warning!

example.com

This site has a poor reputation based on user ratings

View rating details and comments

This site is safe - I want to rate it

example.com
Description:

Popularity:
Ranking: 10 View ranking list

Server location:

See owner at whois Domain Tools
Discover similar sites
Click here if you own this site
Subscribe to this scorecard's comments

Ratings

Reputation ratings are based on real user ratings and they tell you how much other users trust this site.
How reliable are the ratings?

Information from third-party trusted sources

Third-party trusted sources provide additional information from numerous phishing and malware blacklists, and other trusted sources from the web.

Date	Source	Comment
18.11.2012	Symantec	Site has a Symantec Trust Seal. [link]

Comments

Supplement your rating by leaving a comment. Comments provide more information, but do not affect the reputation. Read more

web of trust

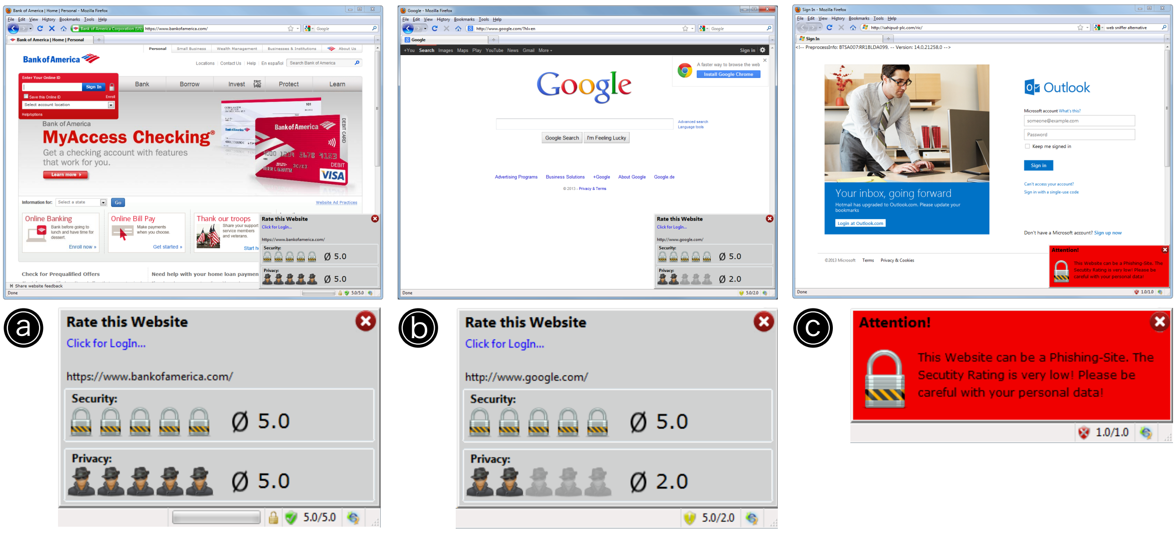
About 170,000,000 results (0.24 seconds)

Safe Browsing Tool | WOT (Web of Trust)
www.mywot.com/
WOT is a community-based, free safe surfing tool fo… es
website ratings & reviews to help web users as they … nline.

WOT - Safe Surfing :: Add-ons for Firefox
https://addons.mozilla.org/en-US/.../wot-safe-browsir…
Web of Trust (WOT) shows you which websites you …
Help support the continued development of W- Safe … mall ...

Chrome Web Store - WOT
https://chrome.google.com/webstore/.../bhmmomiinig…
WOT helps you find trustworthy websites based on n… iences
and is one of Chrome's most popular add-ons.

Figure 5.14: The user intervention components of the Web of Trust extension (a commercial plugin) [322] a) different rating states and the rating icon; b) website indicator and scorecard-popout; c) website alert dialog appearing as a blocking dialog in case of a low rating; d) online scorecard website for each domain; e) integration of web of trust indicators with links on other websites.



Rate this Website
Click for LogIn...
https://www.bankofamerica.com/
Security: Ø 5.0
Privacy: Ø 5.0
5.0/5.0

Rate this Website
Click for LogIn...
http://www.google.com/
Security: Ø 5.0
Privacy: Ø 2.0
5.0/2.0

Attention!
This Website can be a Phishing-Site. The Secutity Rating is very low! Please be careful with your personal data!
1.0/1.0

Figure 5.15: Different examples for dialogs displayed by the community-based rating extension: a) Example for a website rated with a high privacy and a high security rating; b) example for a website rated with a high security but different privacy rating; c) example for a phishing website with very low ratings.

the user for rating the website the dialog contains the URL of the website that the vote will be recorded for and a possibility to open the login screen of our extension.

We had two cases when our plugin did not stay quiet but instead opened a dialog by itself. In case a website had no rating at all we opened the dialog with an additional text explaining that no ratings for this website had been made yet and that the user should please provide new ratings. In case of very low ratings of the website we opened a red warning popup and warned about the low security or privacy rating. The popup did not block interaction with the loaded website but stayed open until it was dismissed. Both kinds of dialogs could be closed using a red cross on the upper right corner of the dialog and did not reappear until the website was loaded again.

Server Side Implementation

On the server side we implemented a few very simple PHP-scripts to build the backend functions that were needed. Using a simple API the web browser extension was able to query ratings for existing websites, could log users in to start a new session (managed by a cookie) and submit own ratings for a user. Whenever a new rating was submitted or updated in the database the new average privacy and security rating was computed and stored. Storing the average values together with URLs allowed us to serve queries for current average ratings faster. Using another query the plugin was able to query the values the user had previously set for a given URL. This was needed to display those values when revisiting a site that was rated earlier. Visiting the backend server as a website new users had the possibility to register accounts by providing a new username and password.

5.3.4 User Study Evaluation

As explained in the beginning our main goal of this project was to determine how well users are able to judge websites in terms of security and privacy and whether such a community-based rating system would be able to alter their judgments or behavior in any way.

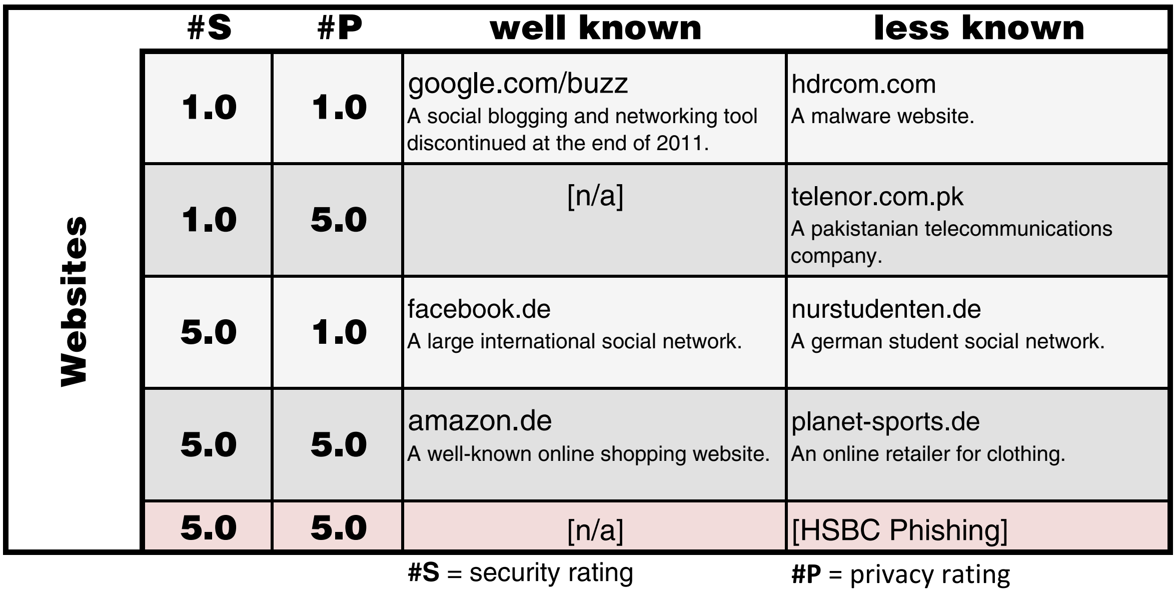
Methodology

We did a laboratory experiment that consisted of three parts. In an opening questionnaire we collected demographic information about the participants. Afterwards the participants had to visit different websites and rate the security and privacy. We concluded the study by debriefing our participants with the details of our browser extension and asked several questions about the plugin concept in general using an exit survey.

We had three independent variables, two as a within-subject factor and one as a between-subject factor and were hence doing a mixed design. Half of our participants had to visit the given websites without having our plugin installed, whereas the other half used a browser that had our plugin installed. As a between-subjects factor the participants had to visit five different websites belonging to different categories. Where applicable we tried to use a well known website and a less known website for each category. For the plugin users we had previously configured all websites to have privacy and security ratings attached to them depending on the context of the website. Table 5.3 shows the different URLs and the assigned ratings for the plugin group. For two categories it was impossible to have a well known website which is why we ended up with a total number of eight websites. We used the alexa.com ratings to find well known and less known websites.

Looking at the categories we used examples for all combinations of high and low privacy and security ratings. As a last category we introduced an error into our given ratings by rating a phishing website with the best security and privacy ratings. We did this to see whether this error would influence the participants behavior.

We counter-balanced the eight different websites that people saw using a Latin-square design. In case of the plugin group we did not introduce our plugin in any way and told the participants of both groups the same story that they would have to judge websites in terms of different factors. When the participants had seen the websites they had to fill in a short questionnaire asking for each website, whether the participants knew the website and whether they had an account with that website. Afterwards we asked six Likert-scale questions to find out more about their security and privacy assessment. Answers to all question should be selected on 5-point scale ranging from '1-low' to '5-high'. The first questions simply asked them to rate the security of the website, followed by a question asking whether the user thought that the website was working correctly and another one that asked whether the website reacts as expected. The next three questions asked for a privacy assessment, whether the website tries to get to know details about the user and whether that could lead to misuse in the future.



	#S	#P	well known	less known
Websites	1.0	1.0	google.com/buzz A social blogging and networking tool discontinued at the end of 2011.	hdrcom.com A malware website.
	1.0	5.0	[n/a]	telenor.com.pk A pakistanian telecommunications company.
	5.0	1.0	facebook.de A large international social network.	nurstudenten.de A german student social network.
	5.0	5.0	amazon.de A well-known online shopping website.	planet-sports.de An online retailer for clothing.
	5.0	5.0	[n/a]	[HSBC Phishing]

#S = security rating **#P** = privacy rating

Table 5.3: The different known and unkown websites that were used for the user study. In some cases a well known website did not exist, reducing the total number of websites that were tested to eight.

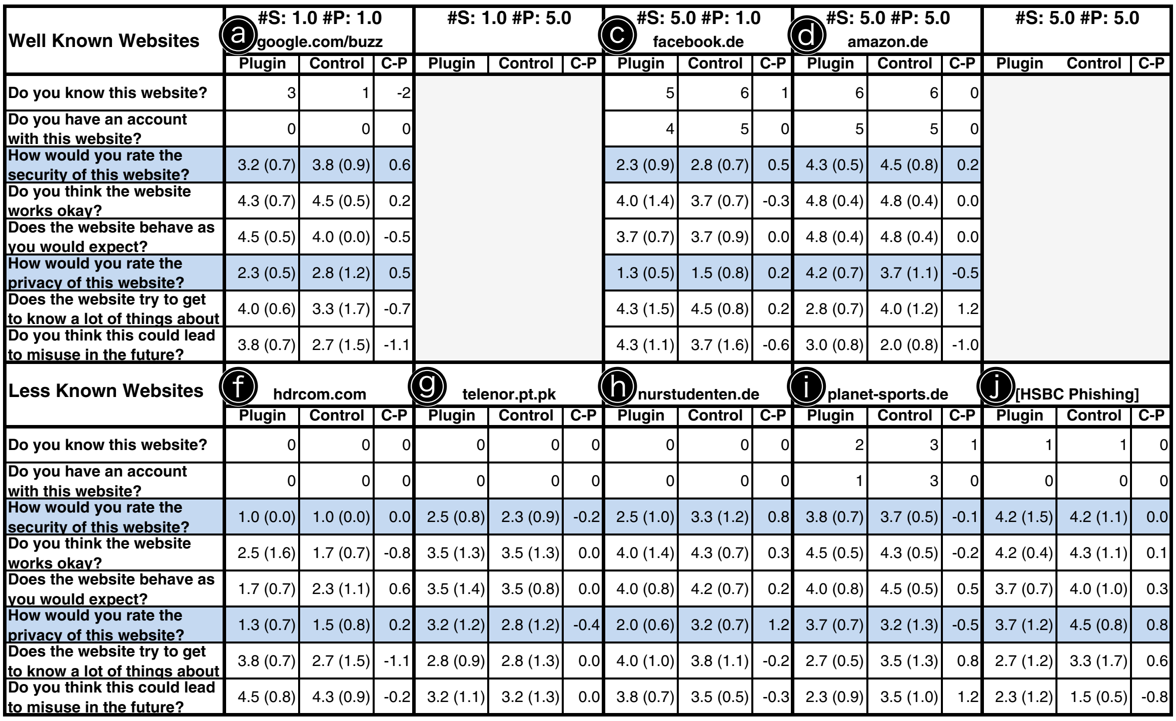
Results

We had 12 participants in our study with an average age of 23 years (range 20 to 28). All participants were male. Most of the participants stated to be rather cautious when using the Internet whilst two people acknowledged to be incautious.

The complete table of all average ratings given by our participants can be found in table 5.4. Looking at the table the first thing one can see is that our classification of known and unknown websites did work reasonably well. Besides “planet-sports.de” all websites were unknown to the participants. And even in this case the website was known to less people than its well known counterpart “amazon.de”. It also seems that our participant samples were similar because the numbers of people that knew the different websites in the plugin and control group are always close to each other.

The most important results to look at are the average answers of our participants to the security ratings and the privacy ratings depending on whether the plugin was installed or not. In table 5.4 these results are highlighted in blue. Using our plugin, a high security or privacy rating of the plugin should possibly raise the respective number in the plugin group. A low number should lower the respective number in the plugin group condition. The second effect should work better, as low scores trigger a warning message, whereas high scores were only displayed in the small indicator in the status bar.

In case of website a) and f) (security 1.0; privacy 1.0) the participants using the plugin indeed rated the security very low. For website a) it was rated 0.6 points lower and for website f) both groups already had the lowest possible rating. Privacy was also rated lower than in



Well Known Websites	a #S: 1.0 #P: 1.0 google.com/buzz			#S: 1.0 #P: 5.0			c #S: 5.0 #P: 1.0 facebook.de			d #S: 5.0 #P: 5.0 amazon.de			#S: 5.0 #P: 5.0		
	Plugin	Control	C-P	Plugin	Control	C-P	Plugin	Control	C-P	Plugin	Control	C-P	Plugin	Control	C-P
Do you know this website?	3	1	-2				5	6	1	6	6	0			
Do you have an account with this website?	0	0	0				4	5	0	5	5	0			
How would you rate the security of this website?	3.2 (0.7)	3.8 (0.9)	0.6				2.3 (0.9)	2.8 (0.7)	0.5	4.3 (0.5)	4.5 (0.8)	0.2			
Do you think the website works okay?	4.3 (0.7)	4.5 (0.5)	0.2				4.0 (1.4)	3.7 (0.7)	-0.3	4.8 (0.4)	4.8 (0.4)	0.0			
Does the website behave as you would expect?	4.5 (0.5)	4.0 (0.0)	-0.5				3.7 (0.7)	3.7 (0.9)	0.0	4.8 (0.4)	4.8 (0.4)	0.0			
How would you rate the privacy of this website?	2.3 (0.5)	2.8 (1.2)	0.5				1.3 (0.5)	1.5 (0.8)	0.2	4.2 (0.7)	3.7 (1.1)	-0.5			
Does the website try to get to know a lot of things about	4.0 (0.6)	3.3 (1.7)	-0.7				4.3 (1.5)	4.5 (0.8)	0.2	2.8 (0.7)	4.0 (1.2)	1.2			
Do you think this could lead to misuse in the future?	3.8 (0.7)	2.7 (1.5)	-1.1				4.3 (1.1)	3.7 (1.6)	-0.6	3.0 (0.8)	2.0 (0.8)	-1.0			

Less Known Websites	f hdrcom.com			g telenor.pt.pk			h nurstudenten.de			i planet-sports.de			j [HSBC Phishing]		
	Plugin	Control	C-P	Plugin	Control	C-P	Plugin	Control	C-P	Plugin	Control	C-P	Plugin	Control	C-P
Do you know this website?	0	0	0	0	0	0	0	0	0	2	3	1	1	1	0
Do you have an account with this website?	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0
How would you rate the security of this website?	1.0 (0.0)	1.0 (0.0)	0.0	2.5 (0.8)	2.3 (0.9)	-0.2	2.5 (1.0)	3.3 (1.2)	0.8	3.8 (0.7)	3.7 (0.5)	-0.1	4.2 (1.5)	4.2 (1.1)	0.0
Do you think the website works okay?	2.5 (1.6)	1.7 (0.7)	-0.8	3.5 (1.3)	3.5 (1.3)	0.0	4.0 (1.4)	4.3 (0.7)	0.3	4.5 (0.5)	4.3 (0.5)	-0.2	4.2 (0.4)	4.3 (1.1)	0.1
Does the website behave as you would expect?	1.7 (0.7)	2.3 (1.1)	0.6	3.5 (1.4)	3.5 (0.8)	0.0	4.0 (0.8)	4.2 (0.7)	0.2	4.0 (0.8)	4.5 (0.5)	0.5	3.7 (0.7)	4.0 (1.0)	0.3
How would you rate the privacy of this website?	1.3 (0.7)	1.5 (0.8)	0.2	3.2 (1.2)	2.8 (1.2)	-0.4	2.0 (0.6)	3.2 (0.7)	1.2	3.7 (0.7)	3.2 (1.3)	-0.5	3.7 (1.2)	4.5 (0.8)	0.8
Does the website try to get to know a lot of things about	3.8 (0.7)	2.7 (1.5)	-1.1	2.8 (0.9)	2.8 (1.3)	0.0	4.0 (1.0)	3.8 (1.1)	-0.2	2.7 (0.5)	3.5 (1.3)	0.8	2.7 (1.2)	3.3 (1.7)	0.6
Do you think this could lead to misuse in the future?	4.5 (0.8)	4.3 (0.9)	-0.2	3.2 (1.1)	3.2 (1.3)	0.0	3.8 (0.7)	3.5 (0.5)	-0.3	2.3 (0.9)	3.5 (1.0)	1.2	2.3 (1.2)	1.5 (0.5)	-0.8

Table 5.4: The answers to the questions of the user-study for the eight different websites. Concerning the first two questions the number of people agreeing with the question was given in other cases the average Likert-scale values and their standard deviations (in parenthesis) are given. The column "C-P" contains the difference between the control and the plugin-condition.

the control group. This may indicate that our warning message did its job in influencing the peoples' ratings. In case of website g) (security 1.0; privacy 5.0) the privacy rating was indeed higher but the security rating was also 0.2 points higher. In case of this website the warning did not seem to have the expected effect. Looking at the websites with a high security (5.0) and low privacy rating (1.0) – websites c) and h) – the privacy ratings of the plugin group were again lower than the ones of the control group. In case of the high security rating the plugin group participants were not influenced in that direction. The appearing privacy warning screen seemed to have had an influence while the positive score displayed alone seemed to have not. In the very positive case (security 5.0; privacy 5.0) three of the four values – websites d) and i) – were higher in the plugin group without any popup appearing. Concerning the last website j) the phishing site was positively rated by both groups. The high security rating that we applied in case of the plugin group had no effect. The privacy was even rated less than in the control group.

Looking at the control group separately one can see how participants would have judged the websites without being influenced. These values show a tendency towards the values that we had previously assigned to the websites. It appears that the less known websites have somewhat lower and somewhat more distributed results. Known brands seem to have higher ratings. However, there are exceptions. “facebook.de” was rated lower than the completely unknown “nurstudenten.de”. Looking at the respective WebOfTrust scores of these websites today facebooks privacy rating is at 66 of 100 whereas “amazon.de” has 93 of 100.

Besides these values our participants also made valuable qualitative comments about the concept. They wanted to see more reasoning for bad ratings which could be easily achieved in a community system by using comments. In case of app stores (like the Android Play Store[15]) this seems to work fine. An overall app rating is always displayed together with comments of the users. Another important comment of the participants was that the terminology of such a plugin should be made more clear, especially if there are multiple types of ratings for one website.

5.3.5 Discussions and Limitations

Looking at the overall results of our study we achieved little to no effect with our given approach. Perhaps the largest effects where the ones when our security and privacy warning were triggered and actively opened a popup. The indicator alone displaying the ratings had no outstanding effect. Since the data can only provide rough trends and are not statistically reliable we omitted a statistical analysis of the data. Nevertheless these results seem to correlate with given findings in other projects and the related work. A small passive indicator is not properly noticed by users. A popup or even a blocking warning work much better.

A major problem of the user study was definitely the small number of participants that we had. With six people for each group (plugin/control) only a single deviating rating can change the average rating a lot. In similar future studies the number of participants would need to be higher. In contrast to a lab study this study could perhaps also be conducted as an online study (see the project in subchapter 5.8 for an example).

Looking at our test conditions we only tested a phishing website with incorrect high scores but not a single one that would have triggered a warning message. We used a malware website for this condition instead. However, these websites can also be identified by the visual look and hence we could not see the potential advantage of our appearing warning.

For our four major conditions we had also taken very extreme security and privacy ratings – min and max values. In a real world setting such ratings would usually not appear. That means that any effect that we would have discovered within this project would be even smaller in a real world setting.

Despite the methodological shortcomings of this evaluation we were still able to see trends of previously already proved results about user intervention and where able to collect valuable qualitative feedback towards a general concept of using community-based ratings for web security.

[15] http://play.google.com

5.3.6 Research Results

Within this subchapter we looked at how people would react to a user intervention mechanism that displays results collected from a community. Although we were not able to measure outstanding changes in the user rating of websites compared to a control group, we received other interesting findings regarding such user intervention mechanisms.

IH *How can HCI be Used to Enhance User Intervention Mechanisms?*

Although recommender systems – for example in online shops – are usually not judged of being a usability or human-computer interaction measure we tried this approach here to make security user intervention mechanisms more usable. When looking at real world computer security problems and how they are used, it is often other – more experienced – people that are asked for advice – usually relatives or close friends. In this context the concept seems to work well, so why not adapt it to the computer to have less experienced people make use of such recommendations, too.

Evaluating such an approach we found two problems: the indicators used where not correctly noticed and perhaps not correctly understood. Other users also missed an important property of inter-personal recommendations within our user interface: personalized comments. Who is the one giving me that recommendation? Can I trust the ratings of a stranger more than the visual look and feel of this website? In an approach focusing on inter-personal relationship these relationships should become clear also by looking at the user interface. The higher the level of importance of the decision is, the more reasonable a rating will need to be.

IM *How can User Intervention be Measured?*

In case of this project we evaluated the user intervention mechanism by collecting Likert-scale answers to different questions regarding websites that we showed the participants. In case of performing such an evaluation it is important to have a control group that uses either an unmodified browser or another user intervention method. This makes it possible to measure the relative performance between both systems. Without such a frame of reference the Likert values are hard to interpret and as these studies are usually conducted within a lab environment the values can't be taken as real world results. The choice between an unmodified browser and a browser using another user intervention concept has to be made depending on which kind of result wants to be achieved. In general, proving a concept superior to the current browser market will always be a first step before comparing a concept against other ones. To elaborate further on this evaluation technique we did a similar evaluation within the project described in subchapter 5.6.

IE *How to Enhance User Intervention Quality?*

In case of this project we were only able to measure small to no effects with our concept. This may be mostly because of the user interface design. As already shown by other related work it seemed that our small indicator showing the average security and privacy scores in the status bar was not noticed enough. The popups seemed to have some influence on the users but they did not contain enough understandable reasons for the negative judgment. In case non-blocking indicators are used they need to be more invasive than small indicator icons in some corner of the browser. In case of project 5.5 we placed our positive and negative feedback within the content area and in case of project 5.6 we used the whole browser chrome as an indicator and evaluated this using the same evaluation technique as within this project. Within this projects we achieved much stronger results.

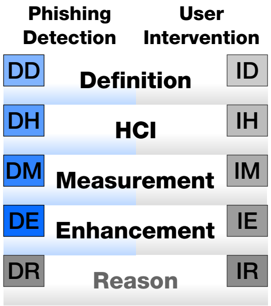
IR *When Should Intervention be Performed to Which Extent?*

This project again showed that in confirmed critical situations a very distinct kind of warning has to be presented to the user – usually a blocking warning. But as errors and other detection problems exist these kind of warnings should be used with extreme care. A question that reappears throughout this thesis is to which extent this rule can be loosened towards non-blocking warnings messages without losing the intervention effect. Small indicators that are aside of the users current primary task are not working but using smaller indicators to reinforce positive situations might also be valuable for several reasons: The user can feel more secure on websites that do not denote a threat and possibly learn to notice the absence of the positive indicators.

5.4 Spell Checking to Detect Fraudulent Websites

This chapter is based on the work that was part of the bachelor thesis "Bulding a Toolkit for Aggregation and Analyzation of Malicious Web Content with Focus on URL Checking" by the student Lukas Höfer [129]. Some parts of the project also led to a publication at the 4th International Symposium on Cyberspace Safety and Security (CSS2012) by Maurer and Höfer titled "Sophisticated Phishers Make More Spelling Mistakes: Using URL Similarity Against Phishing" [186].

So far this thesis discussed new ideas how phishing attacks could be detected, that were derived from user interviews and the users' ideas of security information (section 5.2) and a user intervention method that was derived from social processes for mutual assistance in improving security as it is performed in the real world (section 5.3). In this chapter we use a third approach for coming up with a new phishing detector. We developed a detection mechanism that makes use of a particular property of phishing websites that is crafted by phishers with a certain purpose – in case of this chapter: the URL. Many phishers try to create URLs that look convincing to the average user and hence contain the brand name of the website somewhere within the URL. In other cases they register slightly misspelled domain names that look convincing. Within this project we built a detector that tries to detect those phishing URLs looking for brand names in the URL and for possible typos that can be found using a spell checking mechanism. We also add the quality of a



phishing website as parameter to our later analysis to see whether the phenomenon is dependent on this factor.

We start this chapter by looking at the different parts of a URL more closely and by explaining our algorithms for extraction and analysis of certain parts of the URL. Afterwards we talk about a test set of different URLs and how we ran our algorithms against these. The quality assessment of a sub-sample of our test set is described next before we finally report our results concerning the detection rate and the impact of quality before finally discussing the project findings and its limitations.

5.4.1 Detecting Phishing URLs

Creating a perfect copy of the content area of an original website is an easy thing to do for a phisher. For any website the HTML code and images are all downloaded to the computer of every user of a website and the same code usually can be simply uploaded to another web server to resemble the same website. In contrast to this phishers cannot reuse the exact same domain name as the original website (unless they would have access to DNS servers) and hence have to pick other domains for hosting their attack. The URL of a phishing website thus can be used as one of the best indicators to spot attacks. To create URLs that look at first sight, as if they are connected to the original website, phishers have come up with a variety of different phishing attacks. Among others Krammer et al. [154] compiled a list of such attacks. One of the most sophisticated examples might be the homograph attack [103]. The homograph attack is special case of a character substitution attack that uses different looking letters for domain names. “paypaI.com” might be easily misread as “paypal.com” (on a computer screen) as it is written with a capital ‘i’. The homograph attack itself uses internationalized domain names (IDN) to register domains that look even more close. Some letters from the russian alphabet look exactly the same as their Latin counterparts (russian ‘a’[16]; latin ‘a’[17]). Domain names registered containing the russian version of the letter hence can’t nearly be told apart. Please refer to section 2.3.3 for more URL attacks.

A possible algorithm to find words with nearly identical spelling, is the Levenshtein distance [116] that calculates the number of insertions, deletions and substitutions that are needed to get from one word to another. For our approach we state that unknown domain names that have a close distance to well known domains denote a possible security threat. However, this can’t be applied to a complete URL. So a first questions to solve was which parts of the URL can be used for such checks.

URL Parts and Phishing Examples

A website URL consists of several parts starting with the scheme, followed by the subdomains, the domain name and finally the path section (see figure 5.16). The original URL

[16] http://www.paypal.com (Homograph Attack)

[17] http://www.paypal.com (Original)

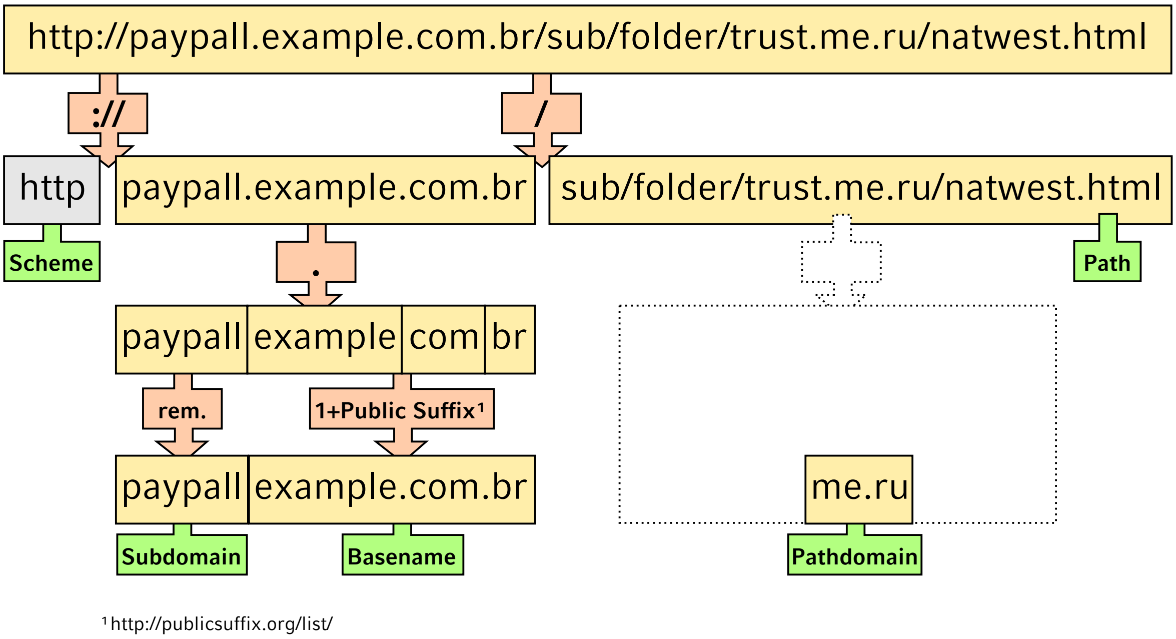


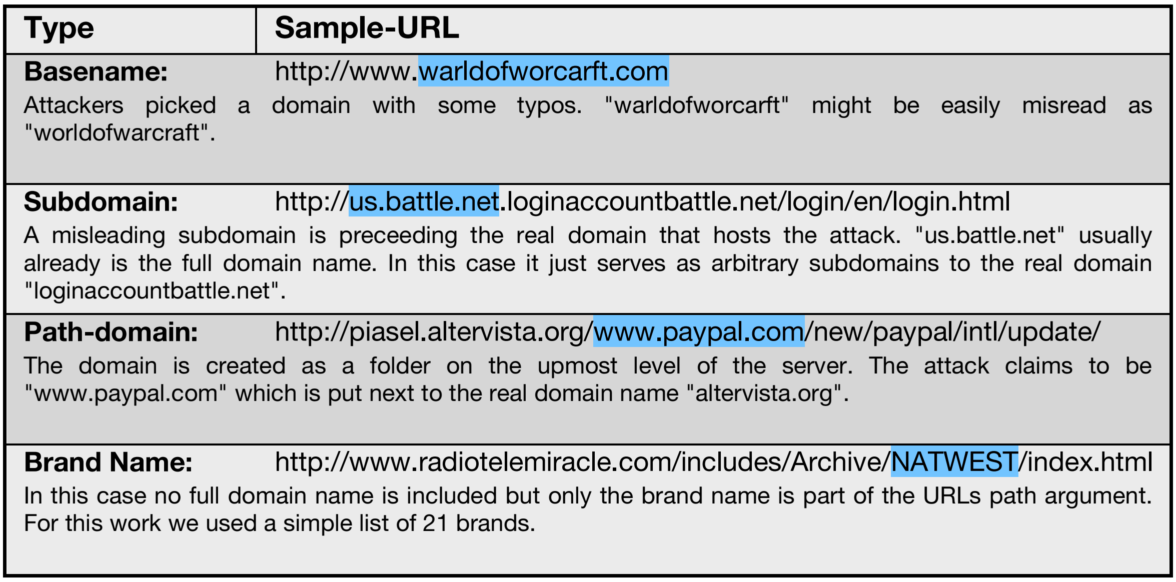
Figure 5.16: Extraction process used to extract relevant URL portions from a URL.

specification [28] contains even more parts like a username and a password for example that haven't been used within our concept because they are very seldom used within phishing URLs. To be able to run a meaningful spell checking query we created an algorithm that extracted the following four types of information from a given URL (see figure 5.16 for the extraction procedure and table 5.5 for four example URLs):

- **Basename:** The basename of a domain is the part of the domain that has to be officially registered with a Network Information Center (NIC) that manages the respective top-level domain. The basename consists of the top-level prefix and the first part before that top-level domain (TLD). In many cases the top-level domain is only one label (e.g. '.com', '.de') but there are NICs that split their own top-level domain again into several so called public suffixes (e.g. '.co.uk'). In such a case the basename has to include the next label of the domain name as well. To be able to properly compute basenames for URLs, a list of the public suffixes is available online[18]. For a phishing attempt the original base domains cannot be used as they are registered by the original companies. This part of the domain is the one where spelling mistakes are most often used.

- **Subdomains:** The rest of domain labels preceeding the basedomain are the subdomains of the URL. Once a phisher owns a domain he can use arbitrary subdomain names. Hence using "us.battle.net" as subdomains for any other domain might fool

[18] http://publicsuffix.org/



Type	Sample-URL
Basename:	http://www.warldofworcarft.com Attackers picked a domain with some typos. "warldofworcarft" might be easily misread as "worldofwarcraft".
Subdomain:	http://us.battle.net.loginaccountbattle.net/login/en/login.html A misleading subdomain is preceeding the real domain that hosts the attack. "us.battle.net" usually already is the full domain name. In this case it just serves as arbitrary subdomains to the real domain "loginaccountbattle.net".
Path-domain:	http://piasel.altervista.org/www.paypal.com/new/paypal/intl/update/ The domain is created as a folder on the upmost level of the server. The attack claims to be "www.paypal.com" which is put next to the real domain name "altervista.org".
Brand Name:	http://www.radiotelemiracle.com/includes/Archive/NATWEST/index.html In this case no full domain name is included but only the brand name is part of the URLs path argument. For this work we used a simple list of 21 brands.

Table 5.5: Four example URLs showing the different types of terms that we extracted from the URLs (based on [186]).

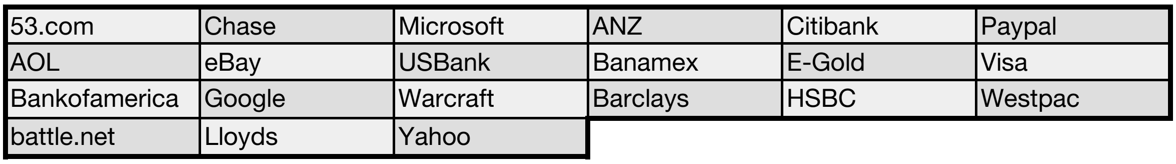
some people in thinking that the actual domain name might be “battle.net” (see table 5.5).

- **Path-domain:** Some other phishers include a faked or original domain name within the path section of a URL. As a path label is usually nothing more than a folder on the respective webspace this kind of attack can even be used if the phisher has no access to the subdomains of a domain (e.g. in case they are using a free web hoster or a IP-adress instead of a domain). We named occurences of domain names within the path portion of the URL “path-domains”.

- **Brand name:** Besides adding complete domain names to subdomains or the path portion many phishers also just place the brand name somewhere to make the URL look more convincing. Such brand names can be easily found without the use of a spell checking algorithm by just searching the URL for the brand name given a list of the attacked brand names exist and that they are distinct enough to be separated from other text.

Extracting URL Parts and Detecting Phishing

The extraction algorithm for the four terms used in our concept was straight forward, as depicted in figure 5.16. We first reduced the whole URL to the scheme, the domain and the path part. After splitting the domain part at each dot it is possible to extract the base domain by finding a valid public suffix at the end of the domain and adding one more preceeding label to it. In case any labels remain, these labels form the subdomains. To find path-domains



53.com	Chase	Microsoft	ANZ	Citibank	Paypal
AOL	eBay	USBank	Banamex	E-Gold	Visa
Bankofamerica	Google	Warcraft	Barclays	HSBC	Westpac
battle.net	Lloyds	Yahoo			

Table 5.6: List of the 21 brand names we used when searching for brand names within a URL. We used brand names that are phished most often (based on [186]).

we applied the basedomain finding algorithm again to the path portion of the URL. To find any brand names we used a list of 21 brand names that are phished most often (see table 5.6) and did a string search for those brand names within the whole URL. We only used such a small number of brand names to see the general effect. In practice, a larger list would be needed.

After having extracted the different terms from the URL, the last important step was to find out whether any of the extracted paths closely resembles a well known domain name. If we had used the Levenshtein distance by ourselves we would have needed a vast number of valid domain names to compare against whilst additionally knowing their importance. Only having those two components it gets possible to compute a similarity score and set a certain threshold that would denote a critical similarity.

We chose not to build such a system by ourselves and instead used the capabilities of a modern search engine that uses exactly such a principle to correct misspellings of their users when entering search terms. So instead of building our own distance measurement tool we submitted the extracted terms one by one to a search engine and checked whether the search engine did return a spell checking result.

5.4.2 Detector Evaluation

To evaluate our detector we wanted to use a large number of different real world phishing URLs to find out how many of the URLs would trigger any spell checking results. As a second research question we wanted to find out whether the perceived quality of a phishing attack correlates with the number of attacks that can be found using such a system. We assumed that phishers that are able to create very similar looking attacks will also spend more effort to make their URLs look similar.

Methodology

We started our evaluation by gathering a test set of 8,730 phishing URLs from phishtank.com. We applied our extraction algorithm to each of the URLs and then submitted queries to a major search engine and checked whether it returned a spell checking result. As brand names did not include any misspellings we just treated the sole existence of a brand

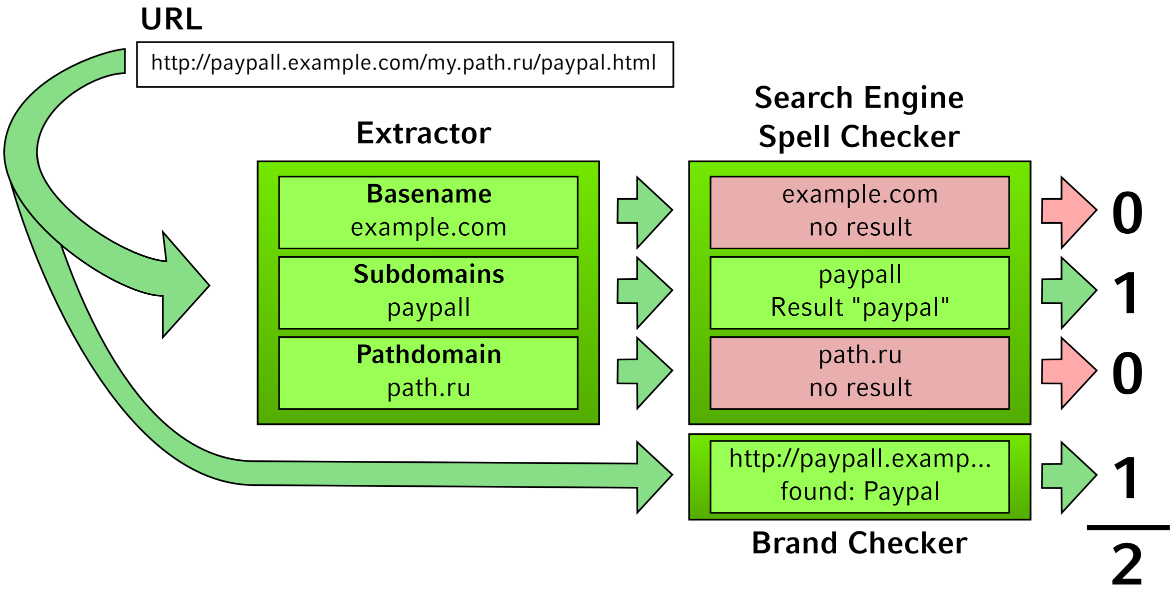


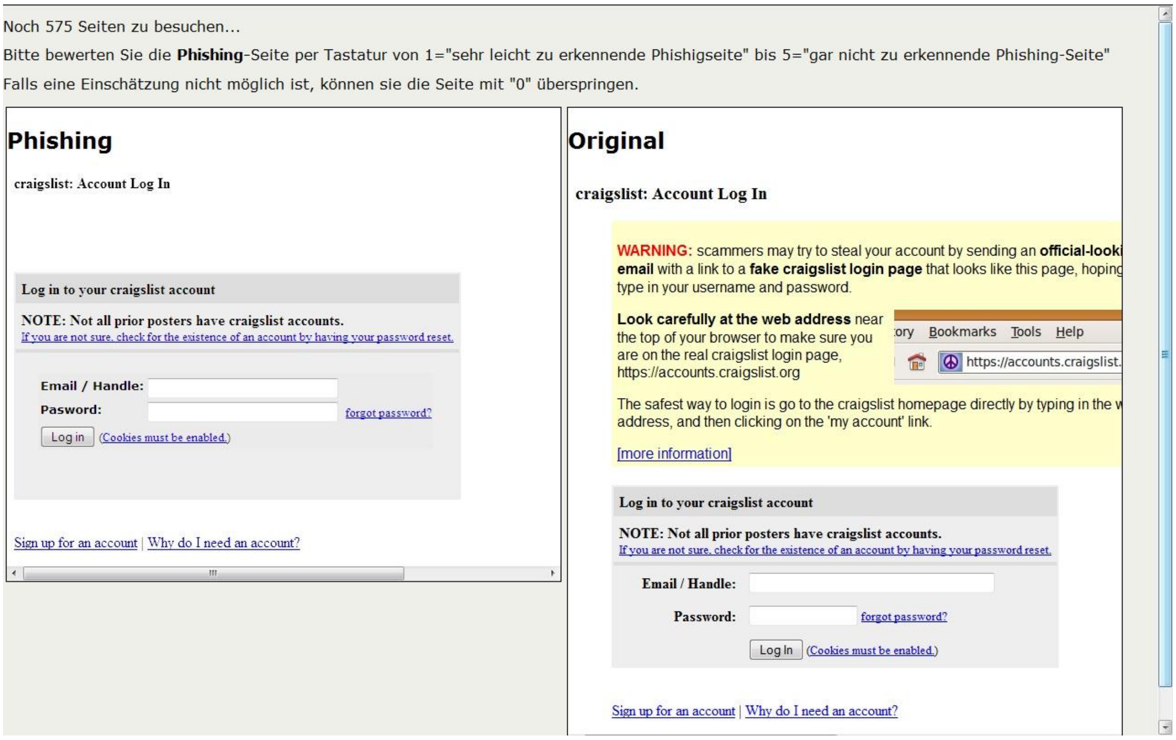
Figure 5.17: Evaluation methodology used to test the domains of our test set. Spell checking and brand name search results were stored for later analysis.

name term as a hit and did not submit those to the search engine. Figure 5.17 shows a diagram of the evaluation methodology. We implemented a Firefox extension to do all the downloading and testing of websites right from the browser.

Quality Assessment

For our second evaluation we used a reduced test set of 566 phishing URLs. These URLs should be rated by experts concerning their visual quality. For those ratings we needed additional information. For each of the 566 phishing URLs we rendered a screenshot of the phishing website and stored it. Afterwards we assigned each of the phishing websites to its original parent website and created screenshots of those 127 resulting parent websites. The whole process is somewhat similar to the process of building the test set presented in subchapter 5.1 but as the large test set was not yet completely ready when this project was carried out we needed to build our own smaller set.

The rating of the websites was performed by three expert Internet users (one IT consultant, one informatics student and one media informatics student). They saw each of the 566 phishing websites in random order next to its original counterpart and used the keyboard keys to judge the quality of the attack from 1-“very easy to recognize the phishing” to 5-“very hard to recognized the phishing” (see figure 5.10 for an example). The experts saw only the browser contents of the websites and did not see any other information (like the URL for example). We did this to avoid biasing of the experts by other factors as we only wanted their assessment of the visual quality of the phish. We did not instruct the experts what factors they should exactly look for and did not define what “very easy to recognize the phishing” would exactly mean.



Noch 575 Seiten zu besuchen...

Bitte bewerten Sie die **Phishing**-Seite per Tastatur von 1="sehr leicht zu erkennende Phishigseite" bis 5="gar nicht zu erkennende Phishing-Seite"

Falls eine Einschätzung nicht möglich ist, können sie die Seite mit "0" überspringen.

Phishing

craigslist: Account Log In

Log in to your craigslist account

NOTE: Not all prior posters have craigslist accounts.
If you are not sure, check for the existence of an account by having your password reset.

Email / Handle:
Pasword: forgot password?
Log in (Cookies must be enabled.)

Sign up for an account | Why do I need an account?

Original

craigslist: Account Log In

WARNING: scammers may try to steal your account by sending an **official-looki** **email** with a link to a **fake craigslist login page** that looks like this page, hoping type in your username and password.

Look carefully at the web address near the top of your browser to make sure you are on the real craigslist login page, https://accounts.craigslist.org

The safest way to login is go to the craigslist homepage directly by typing in the w address, and then clicking on the 'my account' link.

[more information]

Log in to your craigslist account

NOTE: Not all prior posters have craigslist accounts.
If you are not sure, check for the existence of an account by having your password reset.

Email / Handle:
Password: forgot password?
Log In (Cookies must be enabled.)

Sign up for an account | Why do I need an account?

Figure 5.18: Screenshot of the tool the experts used to evaluate the quality of the phishing websites.

False Positives and True Negatives

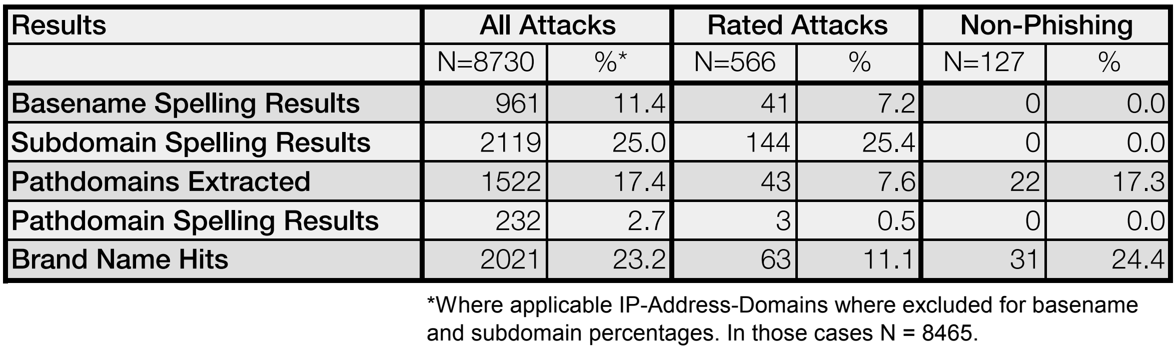
Using the 127 original websites that we had gathered for the quality assessment we could finally also do checks with our algorithm for false positives and true negatives to see whether our detector would bring up a large number of false alarms.

5.4.3 Results

This project produced two different kinds of results, the first group being the detector results for the different terms. We had three different test sets: the whole test set of all 8,730 URLs, the subset used for the quality assessment and finally the set of the 127 original URLs. Afterwards we take a look at how the subset of 566 websites has been rated by the experts and whether the detector results are different for different quality levels.

Detector Results

Table 5.7 shows an overview over the detection results that we achieved with our different extracted terms. Looking at the large result set of 8,730 URLs we had 265 URLs included that used IP-addresses instead of domain names. When calculating the results for domain name based terms, we excluded these domains. For the submitted basedomains the search



Results	All Attacks		Rated Attacks		Non-Phishing	
	N=8730	%*	N=566	%	N=127	%
Basename Spelling Results	961	11.4	41	7.2	0	0.0
Subdomain Spelling Results	2119	25.0	144	25.4	0	0.0
Pathdomains Extracted	1522	17.4	43	7.6	22	17.3
Pathdomain Spelling Results	232	2.7	3	0.5	0	0.0
Brand Name Hits	2021	23.2	63	11.1	31	24.4

*Where applicable IP-Address-Domains where excluded for basename and subdomain percentages. In those cases N = 8465.

Table 5.7: Results for the number of matches for the four different extracted terms (basename, subdomains, path-domain and brand) using three different test sets (full URL set, quality subset and original website set) (based on [186]).

engine returned spelling results for 11.4% of the submitted URLs. When submitting the subdomains we had 25% search engine hits. Looking at the path-domains we were able to extract at least one path domain from 17.4% of the URLs although only 2.7% finally led to a spell checking result. We found at least one of our 21 brand names in 23.2% of all URLs that we tested. In case of path-domains and subdomains the phishers had the possibility of including original domain names that would usually not trigger the spell checker – please see section 5.4.4 for details.

Performing the same checks with our subset of websites we achieved similar results especially for the spell checking of the subdomains (25.4%). Basename (7.2%), path-domain (0.5%) and brand results (11.1%) were a little lower. In general the larger test set should yield more accurate results than the smaller one did.

Our non-phishing test set confirms that our indicators seem to work pretty well. Basename, subdomains and path-domains did not trigger a single spell checking result (0.0%). In 31 cases we found a brand name within the URLs which is perfectly natural as our original URLs certainly contain their own brand names.

Another interesting property of our results is found by combining the different detection methods. The heatmaps in figure 5.19 give an overview over which URL triggered which of the different detection types – each horizontal bar represents one URL of the test set. Looking at the results of the basename, subdomain and path-domain spell checker as well as the brand name detector together 4,552 URLs would have triggered at least one of these features. This means that we achieve a total coverage of 52.1%.

The 52.1% total coverage can be segmented into 43.7% URLs triggering only one detector, 8.0% that triggered two detectors and 0.5% that triggered three different detectors at once. No website triggered all four detectors at the same time.

Quality	very bad [1;2]		bad]2;3]		good]3;4]		very good]4;5]	
	N=226	%	N=149	%	N=112	%	N=79	%
Basename Spelling Results	17	7.5	16	10.7	12	10.7	16	20.3
Subdomain Spelling Results	83	36.7	26	17.4	22	19.6	11	13.9
Pathdomains Extracted	9	4.0	11	7.4	13	11.6	10	12.7
Pathdomain Spelling Results	0	0.0	1	0.7	2	1.8	0	0.0
Brand Name Hits	18	8.0	16	10.7	16	14.3	13	16.5

Table 5.8: The classification results for the subset of the quality test websites clustered by four different quality dimensions (based on [186]) .

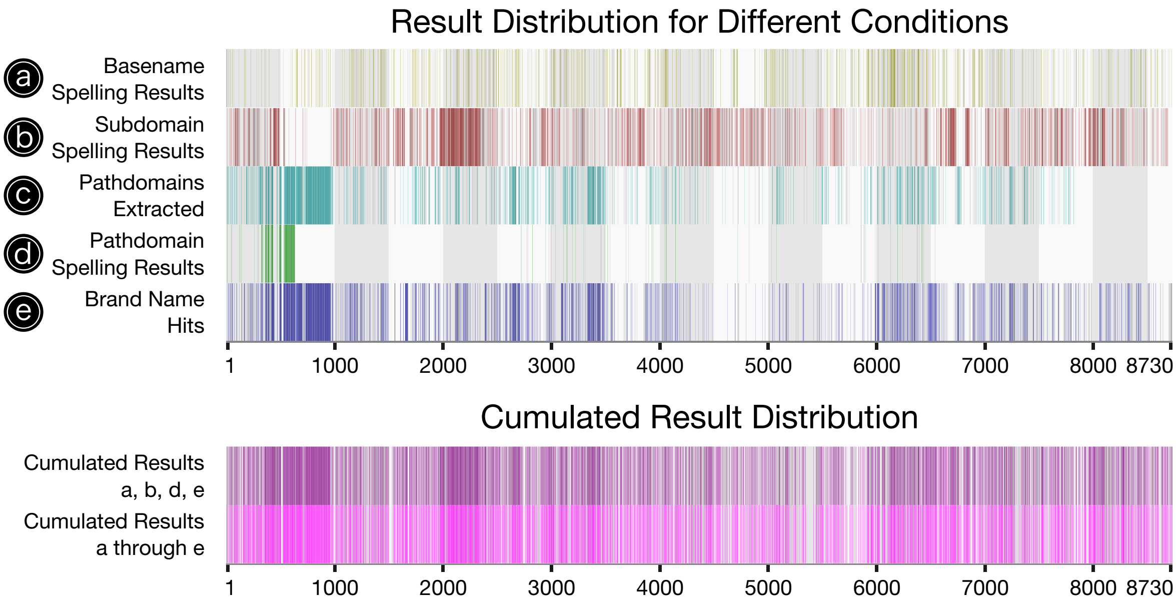


Figure 5.19: Generated heatmaps that show for all 8,730 URLs where which kind of detector or extractor has retrieved a result. The last two rows shows the accumulated heatmaps, with and without taking the extracted path-domains into account (based on [186]).

Quality Results

For each phishing website of the subset of our URLs three experts had rated the phishing quality from 1 to 5. For each website we calculated the average rating and then classified each website into one of four quality categories: very bad (average from 1 to 2 inclusive); bad (2 excl. to 3 incl.); good (3 excl. to 4 incl.); very good (4 excl. to 5 incl.). With this classification most websites were rated very bad (226), 149 fell into the category “bad”; 112 were “good” and 79 websites were classified as being “very good”.

Table 5.8 shows the results of our detector split by the different quality categories. Looking at the percentages one can see that the number of basename spelling results increases from 7.5% to 20.3% with increasing quality. The number of found brand names and the number of extracted path-domains also grows with increasing quality. The only thing that decreases with increasing quality is the number of subdomain spelling results (from 36% to 14%). This is interesting as we saw that this detection rate was the only one being constant between our subset and the whole set of the URLs.

5.4.4 Discussion and Limitations

With this simple URL-based detector we were able to show that a lot of phishing attacks can be detected using our approach when accumulating the different detector values. However, a detection rate of 52.1% is still not enough to have this detector act alone. We would hence recommend to combine the detector with other already existing detection means to further enhance the detection mechanism.

An option to raise the number of detected websites would be to include all URLs that contain a path-domain into the list of suspicious domains. That would have raised the overall detection rate from 52.1% to 54.7% (see figure 5.19). However, this approach would also have a downside: within our non-malicious website test set path-domains have been extracted for 17.3% of the URLs which would hence lead to a lot of false positives.

We think that the detection rate of such a detector could be vastly enhanced. In our approach we only looked at spell checking results for the subdomains and the path-domains. Since the phishers are free to use anything they want within these terms the spellchecker did not necessarily notice the attacks – in case the spelling was correct. We are confident that checking those terms against a whitelist of correctly spelled original domains could find another huge portion of phishing attacks.

When performing the spell checking we just noted whether a correct spelling was suggested or not but did not verify whether the spelling suggestion was really related to the phishing attack or whether it appeared because of a different reason. Our cross check using 127 original domain names shows to some extent that the system does not generate false positives but further testing with a larger sample of original websites would be definitely needed before deploying the concept.

Our quality analysis clearly shows that phishing websites with better quality are a lot easier detected by our approach than websites with bad quality. We argue that when developing a detector, this property should be kept in mind during the whole development process and when possible it should also be tested for, when evaluating the detector.

We also think that the high number of poorly rated website can hence account for the lower detection rates of our subset. When selecting the 566 websites for our subset, it seems that we randomly picked more websites with bad quality then we had in the rest of the subset. This would explain why the detection rates of the subset went down by some percentage points.

Deploying our concept as it was used for the evaluation purposes would create a vast amount of web traffic to search engines to validate the spell checking results. For a field deployment it would be necessary to set up a dedicated service to perform the checks on the different URL parts that would work in a more optimized way.

5.4.5 Research Results

Within this project we built a detector based on observations of technical properties phishers try to tweak to make their attacks more convincing towards the users. In this case we could develop the general detector without turning directly to the end user. This is also the reason why we did not create nor evaluate any user intervention methods for this detection method.

DD *What is Phishing Detection?*

Within this project we introduced "phishing quality" as a factor that we used to measure our phishing detection. This shows that beyond the classic definition of how many websites a detector can correctly classify, there are more dimensions of a phishing attack that could be taken into account and that yield different quality results of a detector. Is each detection of a phishing website worth the same? In our case we were able to show that high quality websites are detected better. Another way of measuring detector performance could also be the amount of money or data that would have been stolen with the detected attacks – although this data is hardly available. This would lead to a different kind of definition of phishing detection. In summary we think, that the number of detected attacks is a reasonable measure for phishing detection whilst other measurements should be performed where possible.

DH *How can HCI be Used to Build Detectors?*

As phishing is a social engineering attack it is not only the researchers that can monitor the users and their security behavior online. In fact the phishers somehow use "usability flaws" to construct and enhance their attacks. Taking this into account one can also learn from them about possibilities where to get hold of their attacks. In case of this project, we observed the tactic that phishers try to make their URLs look convincing. On the one hand this makes the attacks more believable for the user but on the other hand it can be used to track down the attacks using a detector. In those cases the HCI research is more or less done by the attackers and we as researchers only have to pick up their results.

DM *How can Detectors Be Evaluated?*

Besides the classic evaluation strategy of counting the numbers of true and false positives which we also applied within this project, the evaluation of quality as parameter of phishing websites is what was new about the evaluation methodology of this project. Although a tight definition of the quality of a phishing website is hard to come up with, we chose to have that implicitly defined by our experts that were asked to rate the websites. By showing them only the rendered content of the website we somehow already defined the quality ratings as being about visual similarity. Extending the evaluation properties to more fine grained domains can help to get better insights where the detector needs to be enhanced. The large manual workload that came with this type of evaluation can be reduced in case a standardized test set with quality assignments is reused.

DE *What Kind of Detection Works Best?*

The URL as a textual feature seems to have a medium to high potential to build phishing detectors that use these as inputs. A huge pro argument for using URLs as a detector parameter is that they are relatively small pieces of data that can be easily and quickly handled and it is also possible to process large quantities quickly. The downside of any approaches that are based on the URL is that – as other researchers already noted – they are usually not in the focus of the user. That would mean that phishers can change their URL tactics once a URL-based detector is launched to avoid getting detected. The possibility of generating ever new URLs – in the worst case one for each email – is already a huge problem for blacklist based approaches that also work with the URLs. Perhaps a similar problem could arise for other URL-based detectors.

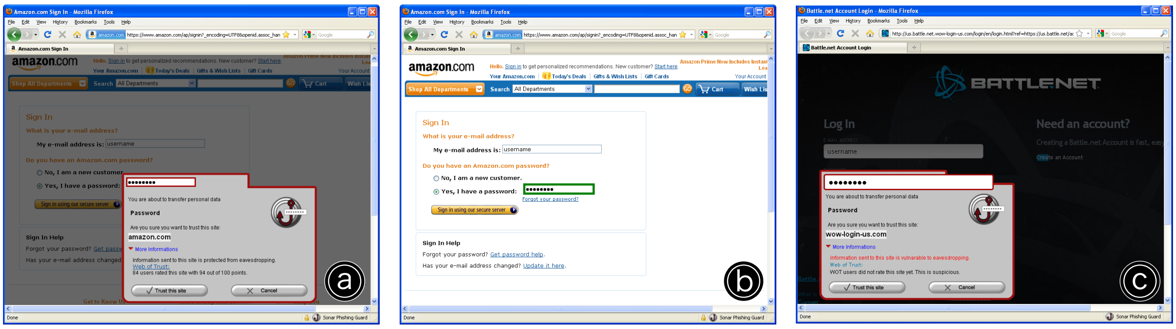
5.4.6 Possible User Intervention for the Approach

Although other researchers showed that domain highlighting in general does not help people to identify phishing attacks [166] we still think that having such a detector to do a first line of detection it would be possible to also create a user intervention interface to present the detection results to the user.

Knowing which term of the URL seems to be problematic would make it possible to attract the users' attention to that specific term and one could even offer an alternative writing or an alternative URL that the user should perhaps choose instead of visiting the malicious URL.

5.5 Data Type Based Security Dialogs

This chapter is based on the work that was part of the bachelor thesis "Keyword Based Security Awareness Warnings for Websites" by the student Florian Müller [201] and the bachelor thesis "Enhancing Datatype Based Security Notifications For Websites" by Sylvia Kempe [150]. Some parts of the project also led to a publication at the 29th international conference on Human factors in computing systems (CHI2011) by Maurer, De Luca and Hussmann titled "Data Type Based Security Alert Dialogs" [182]. A second publication concerning a different part of the project was published at the 7th Symposium on Usable Privacy and Security (SOUPS2011) by Maurer, De Luca and Kempe titled "Using Data Type Based Security Alert Dialogs to Raise Online Security Awareness" [183].



Amazon.com Sign In - Mozilla Firefox

https://www.amazon.com/ap/signin?_encoding=UTF8&openid.assoc_han

Sign In

What is your e-mail address?

My e-mail address is: username

Do you have an Amazon.com password?

○ No, I am a new customer.

● Yes, I have a password: ●●●●●●●●●

You are about to transfer personal data

Password

Are you sure you want to trust this site:

amazon.com

More Informations

Information sent to this site is protected from eavesdropping.

Web of Trust:

84 users rated this site with 94 out of 100 points.

Forgot your password? Get password help.

Has your e-mail address changed? Update it here.

Battle.net Account Login - Mozilla Firefox

http://us.battle.net.wow-login-us.com/login/en/login.html?ref=https://us.battle.net/ac

Log In

E-MAIL ADDRESS: username

Need an account?

Creating a Battle.net Account is fast, easy

You are about to transfer personal data

Password

Are you sure you want to trust this site:

wow-login-us.com

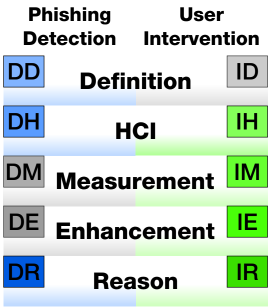
More Informations

Information sent to this site is vulnarable to eavesdropping.

Web of Trust:

WOT users did not rate this site yet. This is suspicious.

Figure 5.20: Screenshots of the different states of the intervention method on real websites: a) password warning (expanded), b) whitelist match on future visits; c) password warning on phishing website (based on [183]).



Reasoning when to display which kind of user intervention has been discussed in the previous chapters and is also a major research question within this thesis. As a security researcher one tends to try to get users as secure as possible without thinking of all the numerous people that will possibly get handicapped by the resulting additions to the workflow during their daily work routines. When tackling this from an HCI perspective one has to think of both sides and has to notice that a general anger of the users towards security measures will never aid in better security behavior.

In this chapter we want to introduce a way between non-blocking indicators and annoying blocking dialogs (e.g. figure 4.1). We call them "semi-blocking dialogs". We based this project on the assumption that not every data that is entered on a website is so critical that it might get stolen (although the warning in figure 4.1 suggested that). In fact we assume that only certain kinds of entered data (e.g. credit card data, login data) is what needs to stay protected. A phisher that gets hold of a search term a user submitted to a search engine will usually not be very happy. Using those specific "data types" – as we will call them throughout this subchapter – it is possible to postpone security checks and more importantly security intervention to the moment when critical data is involved. Using this concept we were able to create a user intervention mechanism that appears in the context of critical data right in the user's focus (see figure 5.20).

Within this chapter we will first describe the general concept of our dialogs in more detail before moving on to a first prototype and its lab evaluation that we did. As a follow up to this first prototype we refined the warning design using a focus group and tested it again in a field and a second lab study. For more details about this project please compare the publications associated with this chapter [182, 183].

5.5.1 User Intervention Concept

To fully explain the user intervention concept of this project we will first start explaining the behavior towards the user with a practical example before turning to the three main contributions made by this new kind of user intervention method: Taking the users' situation into account; using semi-blocking dialogs; in-context dialog appearance.

The User Perspective: Alice Does Online Shopping

Alice is about to do online shopping on a new shopping website that she has been invited to by a good friend Bob. She visits the website, composes her order and moves on to the checkout phase. When she is about to enter her credit card data, a warning dialog appears surrounding the input field of her credit card number (see figure 5.20). At first she didn't even notice it, as she was still looking at her keyboard while entering the credit card digits. Now that she looks up to the screen again she sees a warning telling her that she has never entered credit card data on that website before. The warning also contains information about the fact that the data she just enters won't be protected from eavesdropping and that the website is poorly rated by other people. That sounds weird and she calls her friend Bob that was supposed to have invited her. Bob has never heard of such an email but tells her that he caught a computer virus a few days ago. Luckily Alice did not enter and submit all her data on that fraudulent website.

A few days later Alice decides to buy something at the well known online retailer she knows for years. As she opens the website and starts entering her password the password field turns green. "At least here everything is fine" shes says and enjoys the rest of her online shopping.

The General Concept

Within our concept we introduce two different kinds of user feedback the choice of which is dependent on the type of data a user enters into a form field. Whenever our system detects that the user enters such a critical type of data into a form field we check whether this website's domain is contained in the user's personal whitelist – whether it has been acknowledged before. So far our prototype checks for three different kinds of data: credit card numbers, passwords and bank transaction numbers (one time passwords used for bank transactions in Europe). In case the website has been previously acknowledged by the user the respective form field is highlighted with a green border indicating that the plugin spotted the input of critical data but acknowledges it. In any other case a user intervention window opens up that displays details about the current website and the type of data that was recognized. While the warning is open the user can still continue typing text into the respective form field but the access to other form fields and the form submission is blocked. The user can now acknowledge this website and add it to the personal whitelist or alternatively cancel the input and leave the website.

Compared to the previously presented concept this concept does not involve a detector that tries to detect phishing sites by looking at different technical properties. A technical detection is needed to find the different data types but besides checking the whitelist no website is judged as being phishing or not (see section 5.5.9).

Overall this general user intervention concept introduces three new user intervention properties.

Situation Specific Warnings

Warning concepts that have been presented so far all rely on technical properties to assess a certain risk level before displaying an actual warning. In this project we use context information of the action that the user is about to perform. In our case we use the type of data that the user is about to loose to be able to know when the user is entering a critical context state that needs additional security attention. Any appearing critical data type hence raises the security importance. This approach could be coupled with a detector which is just left out in our project here. Other situation information could be used to lower the security importance (e.g. if the user does not have an account with a brand that appears on a phishing site).

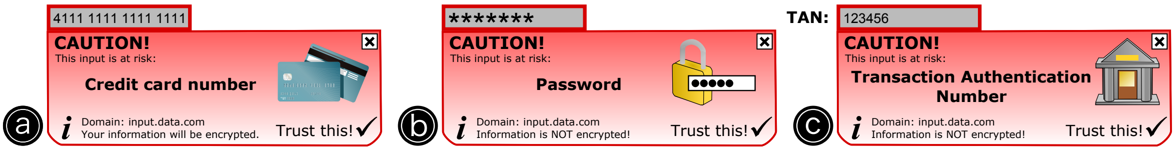
Reducing the number of warnings based on the situations has the major advantage that it can avoid habituation towards a warning message. Another advantage of using the user's situation before displaying a warning is that the user intervention methods now can incorporate that situation to explain to the user why the warning appeared. In our case the warning clearly states that the user is about to enter a credit card number for example.

Semi-Blocking Dialog

Related Work and the last subchapters showed that it is extremely hard to gather the user's attention with non-blocking warnings. When we started developing this concept we originally thought of a blocking warning message (that immediately blocks access to that website as soon as a data type is detected). In practice this has the problem that it interrupts the user's course of interaction so heavily that it might confuse the user – especially if the warning has appeared in error. We experienced this when the blocking warning appeared while people were typing, they kept typing without noticing the warning. Most of their password input was then lost due to the changing window focus and in case the warning appeared at an original website they had to retype their password. This is why we came up with the semi-blocking dialog. It appears and blocks the interaction with most of the existing website but leaves the interaction with the current form field possible. This way, the user can continue typing and can then verify the warning message when her focus switches back to the computer.

In-Context Appearance

A third novelty that we added to our concept here, was to have the warning messages appear in-context to the user's current focus. In case of input fields this is easy to determine as the users' focus usually is where they are currently typing. This in-context appearance has advantages in several dimensions.



4111 1111 1111 1111

CAUTION!
This input is at risk:
Credit card number
Domain: input.data.com
Your information will be encrypted.
Trust this!

****

CAUTION!
This input is at risk:
Password
Domain: input.data.com
Information is NOT encrypted!
Trust this!

TAN: 123456

CAUTION!
This input is at risk:
Transaction Authentication Number
Domain: input.data.com
Information is NOT encrypted!
Trust this!

Figure 5.21: The three different data type based warnings used with the first prototype (based on [183] and staged for printing).

Location: Using the user's input text field as the assumed visual focus area of the user makes it possible to make the warning appear right in the user's view. Even if she looks away from the screen when the dialog appears she will look back to the input field when done typing.

Timing: Making the warning dialog appear together with the user's most critical action – right in the moment when critical data is entered – could greatly help the user to understand why the warning has just appeared. It couples the dangerous action and the warning more closely than other user intervention procedures do.

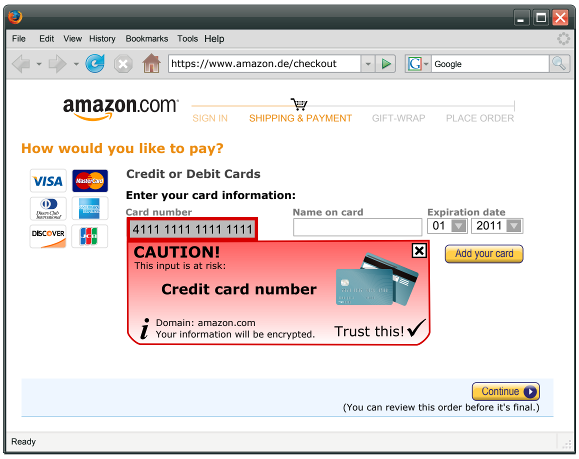
Preserve Context: Our semi-blocking warning appears at the user's focus and hence can also be more moderate in size and have a reduced amount of alertness than other non-blocking or blocking warnings do. Most of today's browser warnings fill at least the whole content area of the browser. This makes it harder for the user to understand what the current warning is about. In case of our warning it is integrated right at the user's focus and the rest of the previous interaction area is still visible preserving the cognitive context that the user currently is in.

5.5.2 The First Prototype

Both prototypes that we developed throughout this project were created for the Mozilla Firefox browser. A screenshot of the design of this first indicator can be found in figure 5.22. We designed the dialog to contain the type of data as the most prominent information. Each data type is represented by a different icon and is additionally labeled with a text using a large font. An info section in the lower part of the dialog shows additional information by repeating the domain of the website that the user currently visits and by displaying the encryption status. Using a button labeled "Trust this!" the user can add this website to her whitelist or close the dialog using a small "X" in the upper right corner.

5.5.3 Detecting the Data Types

As mentioned earlier we implemented detection for three different types of data which each needed slightly different algorithm for the detection (see figure 5.21 for dialog examples):



How would you like to pay?

Credit or Debit Cards

Enter your card information:

Card number: 4111 1111 1111 1111

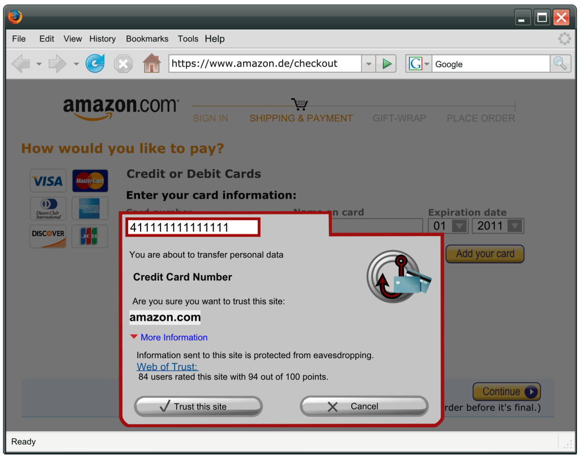
Name on card

Expiration date: 01 2011

CAUTION!
This input is at risk:
Credit card number
Domain: amazon.com
Your information will be encrypted.
Trust this! ✓

(You can review this order before it's final.)

Figure 5.22: User interface of the first data type based warning after the user has entered a credit card number on an unknown website [182]. [staged for printing purposes]



How would you like to pay?

Credit or Debit Cards

Enter your card information:

4111111111111111

Expiration date: 01 2011

You are about to transfer personal data

Credit Card Number

Are you sure you want to trust this site:

amazon.com

More Information

Information sent to this site is protected from eavesdropping.

Web of Trust:

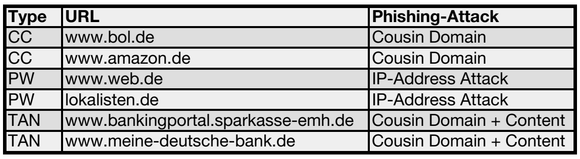
84 users rated this site with 94 out of 100 points.

✓ Trust this site ✕ Cancel

rder before it's final.)

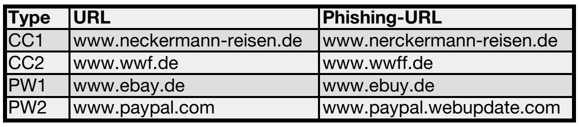
Figure 5.23: Final user interface for the data type based user intervention method that was used for the second user studies [183]. [staged for printing purposes]

- **Passwords:** As password fields are a special kind of input field in a HTML website we just look for those being of the type "password" and trigger our warning whenever the first character was entered in such a field.
- **Credit Card Numbers:** Most credit card numbers can be easily verified using the LUHN algorithm [283]. The algorithm is a special sum over all digits of the credit card. Starting from the end every second digit is doubled before adding it to the sum and in case the multiplication result is greater or equal to 10 the number 9 is subtracted again. If the sum of all digits ends on zero the LUHN algorithm has verified the code. For "2758" the sum would hence be $8+(5*2-9)+7+(2*2)=20$ which would be a valid LUHN code. Since this cannot be a credit card number by itself we also introduced a length check.
- **TAN-Numbers:** TANs are short one time passwords used for bank transactions in Europe. They are usually 4 to 6 characters long and mostly consist of digits only. This definition clashes with a lot of other input types (e.g. postal codes). To avoid getting warning messages appearing in error we included a HTML-search algorithm that looks for matching keywords in the HTML-DOM-Tree vicinity of the input field. In this first implementation this algorithm can be easily fooled by an attacker in case he replaces the textual labels using images.



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 5.9: The 12 websites used for the preliminary lab study. In case of the phishing websites we used the given attack to derive a phishing website URL from the original URL (based on [183]).



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 5.10: The websites used for the second lab study (based on [183]).

5.5.4 Lab Evaluation

To test the preliminary design of our user intervention method we performed a lab study to find out how many participants could be protected from phishing using our plugin compared to a standard browser.

Methodology

We used a mixed-model design for that lab-study having the use of the plugin (with plugin/without plugin) as a between-groups independent variable. As one within-subjects factor we wanted to use the three different data types that our plugin was able to detect and as a second within-subject factor we wanted to see how people would behave on original and on malicious websites. Therefore we needed six websites that we could show each participant. To be able to balance the study we hence needed a total of twelve websites (six original and six malicious) (see table 5.9).

For our study we developed a scenario that we called “Grandma is Ill” to guide the user through our study tasks. In case of security lab studies it is usually impossible to have the participants use their own data and instead, role playing is often used to have the participants carry out the tasks as if they were someone else. This is often criticized as the participants may not take the tasks of the user study seriously. In our approach the user presumes performing tasks for his own grandmother and hence isn’t really role playing but just carrying out tasks for another person using their security details. I compare different scenario possibilities and the advantages of this approach in more detail in section 7.2.1.

Using the scenario we sent the participants to six different websites – three malicious (phishing) and three original ones – and measured the number of malicious websites that they would refuse (true positives) and the number of original websites that they would actually use (true negatives). Our dependent variable hence was the “correctness” of their decisions.

For none of the websites we had our participants use an actual online website instead we diverted all the traffic back to the local computer using the windows “hosts” [304] file and served identically looking copies of the websites from there. It was impossible for the participants to notice that the traffic did not come from the Internet and the URLs they saw all looked like the original ones. Please see the project in subchapter 5.9 for more technical information and the best way to conduct such a study.

Looking at the websites in table 5.9 again we used shopping websites for the credit card condition, banking websites for the TAN condition and a social community and a webmail service for the password condition. For each condition we presented the user with a short text explaining why he had to visit the website for his grandmother and what she wanted the participant to do there. Grandma had all links bookmarked for the participants in her browser and they also received her “secret notebook” that contained her passwords, TANs and credit card information.

We balanced our different conditions using a 6x6 Latin square which was used twice whilst inverting the phishing websites in the second set. These twelve different sequences were then used for both between-subject groups (with plugin/without plugin).

We instructed our participants to “think aloud” during the study and did not mention security until the final debriefing. In case a participant had concerns about entering information on a website we told them that they are allowed to skip tasks if they fear any negative consequences for their grandma. After all six tasks the participants were debriefed and had to fill out an exit survey containing also qualitative questions about our concept.

Results

As already mentioned we had 24 participants in our study most of them being students as they were recruited around campus. Each participant was randomly assigned to the between-subjects groups as long as space was available. In the plugin group the participants were on average 24 years old (three being female) whilst in the control group we had an average age of 23 years and three female participants.

Looking at our quantitative results our subjects using the plugin refused to enter data on 20 of the 36 appearing phishing websites (55%) whilst in the control group only five phishing websites were detected (13.9%). Analyzing the results statistically using a two-way mixed ANOVA – see section 7.2 for more information about statistical tests – showed a significant main effect for our between-subjects factor ($F_{1,22} = 11.83 p < .05$). The within-subjects factor “data type” (credit card, password, TAN) was not significant ($F_{2,44} = 0.77, p > .05$) but we had an interaction effect of the two independent variables ($F_{2,44} = 6.27, p = 0.004$). Plugin combined with the data types for credit card and password did not show significant changes in recognition but the differences between the TAN and password recognition was significant ($F_{1,22} = 6.50, p < .05$). The reason for this most probably is that both groups found four phishing websites in the TAN condition whereas our plugin seemed to have a lot more impact for the other two data types.

The participants of the plugin group generated two false positives by refusing to enter data on original websites whereas no false positives appeared within the control group (without the plugin).

In the qualitative survey part of the study we explained the concept to all our participants and they rated the helpfulness of the plugin on a Likert scale from 1-‘not helpful at all’ to 5-‘very helpful’ positively with a median of 4. They liked the opportunity to think again before submitting critical data but many people felt that the warning screens that popped up on every site were annoying. This was actually a problem of the study setup as we tested six websites and a warning appeared for each one of them. In practice warnings would show up more seldom (see the upcoming field study). Some users also complained about the user interface of the warning which is why we chose to redesign it in a second iteration.

5.5.5 The Second Prototype

After testing our first prototype in the lab study we had identified several design flaws. The most importants one being that we had a very big “Trust this!” button compared to the small “X” that somehow suggested to the user to choose the unsafe option as a standard. This should not be done in security dialogs. As a second implication the wording of our dialog was still very technical using terms like “encryption”. To solve these issues we wanted to redesign the dialog based on the findings of a focus group with five participants (mostly students with an HCI background).

Before the focus group we created a few new design iterations by ourselves but kept them hidden from the participants until the end of the focus group (see figure 5.24 f through h). Within three phases we wanted to examine how a well working dialog needs to look like. In the first phase we explained the concept of data type based warnings to them but without showing any warning imagery. Afterwards we orally discussed different design properties (e.g. colors, graphics, headlines) with the participants. In the second phase we had the participants craft their own dialogs with pencil and paper. We also provided additional material to each participant – like an assortment of graphics and icons or various colored background drops. In the last phase we discussed the drafts with the participants and also showed them the drafts we had previously created. We asked them to vote for the ones they liked best and to explain why they chose the respective design. Figure 5.24 shows the dialog drafts created by the participants and the ones created by us. Talking about all final drafts the participants did not want a “X” style button to close the dialog as well as they wanted that the URL of the website that is visited is most prominent in the dialog. They also proposed an area with more security information that could be considered when necessary. Looking at the drafts in figure 5.24 they liked several drafts very much (b and f).

We not only incorporated several of the enhancements found through the focus group but also added some more technical security to the plugin. In case form fields were filled by the autocomplete machanism of the browser our plugin was so far not yet triggered. We corrected this for the second version. The final dialog design can be found in figure 5.23.

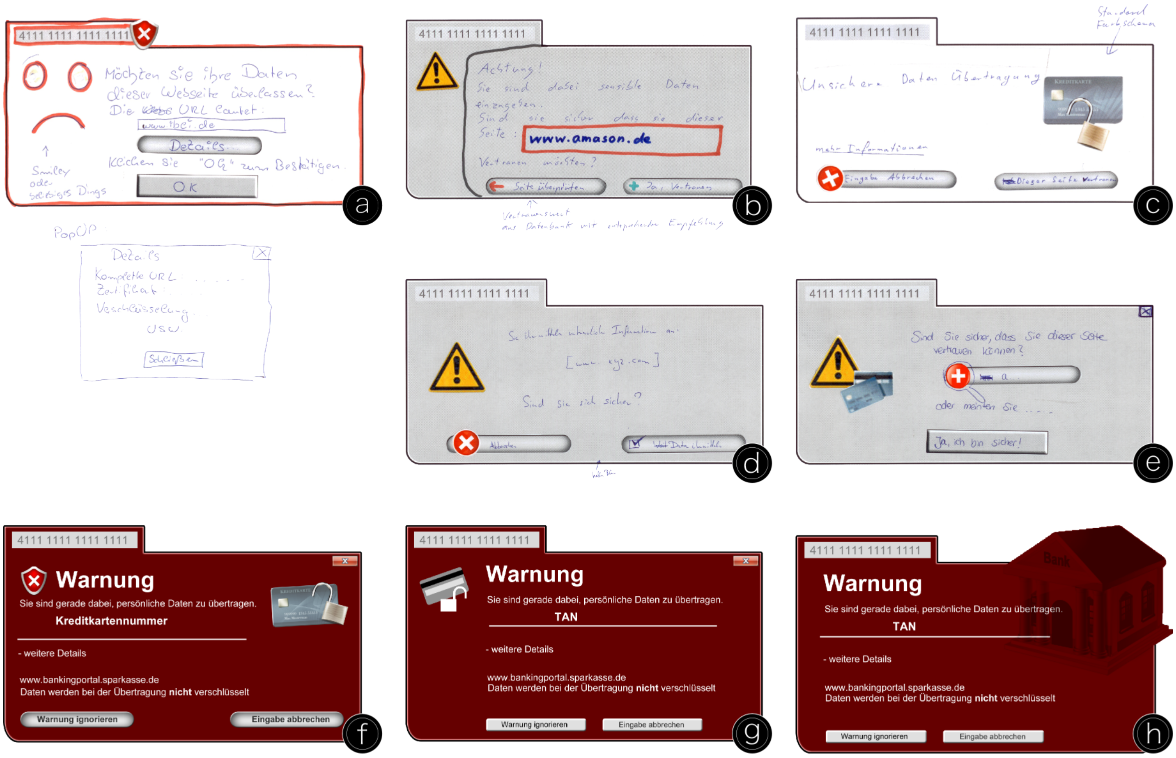

a
4111 1111 1111 1111
Möchten Sie Ihre Daten dieser Webseite überlassen?
Die URL lautet:
www.ibe".de
Details
Smiley oder sonstiges Ding
klicken Sie "OK" zum Bestätigen
OK
PopUp
Details
Komplette URL
Zertifikat
Verschlüsselung
usw.
Schließen
b
4111 1111 1111 1111
Achtung!
Sie sind dabei sensible Daten einzugeben.
Sind Sie sicher, dass Sie dieser Seite: www.amason.de
Vertrauen möchten?
Seite überprüfen
Ja, Vertrauen
Vertrauenswert? aus Datenbank mit entsprechender Empfehlung
c
4111 1111 1111 1111
Standard Farbschema
Unsichere Daten Übertragung
mehr Informationen
Eingabe Abbrechen
Dieser Seite Vertrauen
d
4111 1111 1111 1111
Sie übermitteln sensible Informationen an [www.xyz.com]
Sind Sie sich sicher?
Abbrechen
Weiter Daten übermitteln
e
4111 1111 1111 1111
Sind Sie sicher, dass Sie dieser Seite vertrauen können?
oder meinten Sie ...
Ja, ich bin sicher!
f
4111 1111 1111 1111
Warnung
Sie sind gerade dabei, persönliche Daten zu übertragen.
Kreditkartennummer
- weitere Details
www.bankingportal.sparkasse.de
Daten werden bei der Übertragung nicht verschlüsselt
Warnung ignorieren
Eingabe abbrechen
g
4111 1111 1111 1111
Warnung
Sie sind gerade dabei, persönliche Daten zu übertragen.
TAN
- weitere Details
www.bankingportal.sparkasse.de
Daten werden bei der Übertragung nicht verschlüsselt
Warnung ignorieren
Eingabe abbrechen
h
4111 1111 1111 1111
Warnung
Sie sind gerade dabei, persönliche Daten zu übertragen.
TAN
- weitere Details
www.bankingportal.sparkasse.de
Daten werden bei der Übertragung nicht verschlüsselt
Warnung ignorieren
Eingabe abbrechen

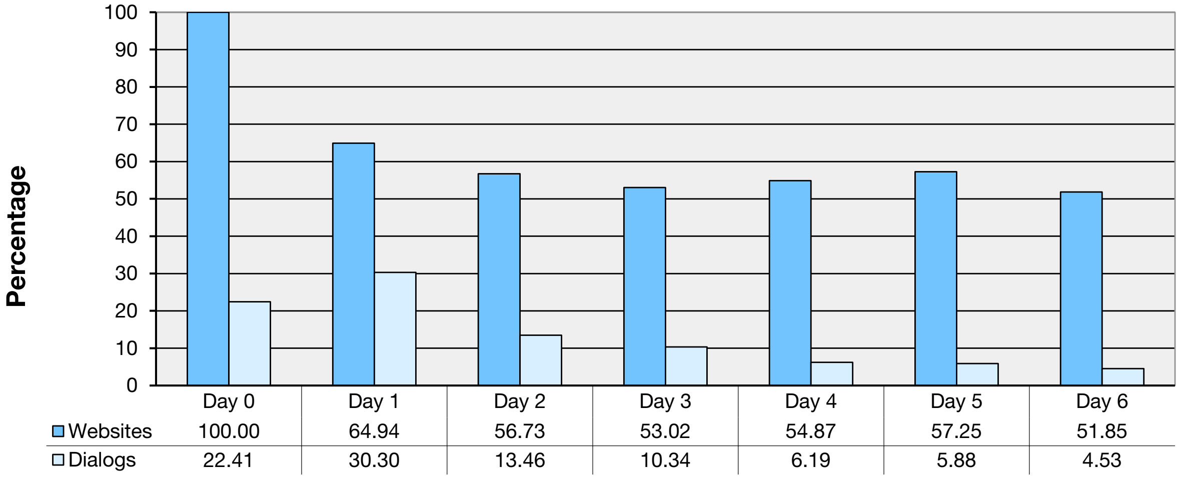
Figure 5.24: Design drafts for the second prototype created by the participants during the focus group (a through e) and beforehand (f through h) (based on [183]).

5.5.6 Field Evaluation

To evaluate the enhanced plugin prototype we conducted a field study having participants use the plugin at home. We did this to collect information about the real world plugin usage (by quantitative and qualitative means). For the quantitative evaluation we logged certain pieces of usage information and for the qualitative evaluation we sent an online post study questionnaire to our participants.

Our participants could download the plugin from a public download website that we had set up. After installing the plugin it asked for the users' email address to be able to send them the post study questionnaire.

For our data collection process we wanted to find out how the plugin integrated in the users' daily web usage. Therefore we needed to log their online browsing behavior to some extent. Logging the actually visited websites would have denoted a privacy issue towards the participants. As a solution we hashed the visited domains before transmitting them to our server. This allowed us to recognize whenever a website was visited a second time or was reused between participants but we were not able to reverse the process to find out which URL was actually visited. Besides the visit data we also collected data whenever a plugin warning appeared or the user interacted with the warning dialog. Besides our anonymization process



	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
🟦 Websites	100.00	64.94	56.73	53.02	54.87	57.25	51.85
□ Dialogs	22.41	30.30	13.46	10.34	6.19	5.88	4.53

Figure 5.25: Percentage of new websites and and appearing warning dialogs throughout the seven days of usage [183].

we did not link the collected data in any way to personal data of the participants. Instead we only stored an anonymized participant id.

14 participants from 22 to 68 years (avg. 40 years) used our plugin for a period of more then seven days. For all participants we only used data entries from the point of first usage upto exactly seven days later. The average Internet experience of our participants – rated from 1-“not at all experienced” to 5-“very experienced” was 4.2 (SD 0.97).

Using the collected web usage data we were able to find out, which percentage of the websites the users visited each day had never been visited before. Using this data together with the number of appearing dialogs we also could evaluate for which percentage of the websites a warning dialog appeared. Figure 5.25 shows these percentages. Within the first 24 hours naturally all websites had not been visited before. Afterwards the number of new websites drops quickly to an average level of about 50% and stayed like this for the forthcoming days. In contrast to that the number of appearing warning dialogs of our plugin shows a continuous downward trend after the first day.

We actually assumed that both measurements would continue to decrease over time, but it seems like our users kept visiting a certain amount of new websites each day. However, the number of dialogs that appeared still decreased more and more to a minimum of 4.5% on the last day of our experiment. This shows that although users visit a lot of new websites every day the number of new websites that involve critical data seems to drop proving that our concept even without a detector would work out after a certain amount of training time. In total, 229 dialogs appeared to our participants asking them for their decision about that website. 112 times that website was added the user’s whitelist and 32 times we logged an actual dismissing of the dialog. As we were missing logging data for the remaining 85 cases this means that the users must have navigated away from the website without taking any decision at all. In 522 cases a website visit created a hit on the user’s whitelist and hence a positively highlighted input field appeared. Concerning the 229 dialogs that appeared the "more information" section of the dialogs was only unfolded 11 times – by only 4 users. This is highly problematic as it contains important information about security details of the website (please refer to the discussion section).

From the post study questionnaire we drew other interesting findings. None of the participants reported to have fallen for a phishing attack. They rated their phishing knowledge with 3.1 on a Likert scale from 1-"I don't know anything about it" to 5-"I know it very well" but stated to care a lot for the online security (average 4.4). The general concept of the plugin received an average score of 4.3 and usage was rated with 3.9 in average. Again this lower score might be due to the fact that users reported that in several cases the plugin detected a wrong data type. Although still a lot of warnings opened up in our field trial, the participants experienced that the number of warnings got less. On a Likert scale from 1-"the warnings did not get less" to 5-"the warnings got less quickly" they answered with a median score of 4.5. The coloring of the input fields and the positively enhanced green indicator around input fields was especially liked by the participants.

5.5.7 Second Lab Evaluation

Besides the field study we also performed a second lab study with the second prototype that was very similar to the first user study. In this second study we limited the amount of data types tested to password and credit card number (excluding bank transaction numbers) (see table 5.10). We had 16 participants with an age ranging from 19 to 51 (average 28) – four were female. We ensured that no one had taken part in any of the previous studies. To balance the expert level of the users in our two groups we tried to assess the users' security knowledge before starting the actual study without actually asking for security knowledge to avoid priming. We hence asked them to rate the "understanding of Internet technologies" from 1-"no knowledge" to 5-"very good knowledge". People with an answer of 4 or 5 were considered as experts and equally split among the conditions. In the end we had one more 'non-expert' in the plugin condition than in the control condition.

In the second lab study both groups discovered more phishing websites than in our preliminary study. The plugin group found 12 of the 16 phishing attacks (75%) compared to 44% in the control group. Although this looks like a huge difference we found no statistical difference in this study ($F_{1,14} = 2.01; p = .187$) – mixed ANOVA on the variable plugin usage (yes/no). In the plugin group we had one false positive. The fact that we found no statistical significance in this study might be due to the fact that all but three participants had been classified as experts and hence the plugin was less useful to them than to a higher amount of non-experts in the first lab study. After the security related part of the study we asked the people more directly for the phishing knowledge and found that our plugin group had only a median of 3 whilst the control group had a median of 4. This could also be a reason why we did not see any statistical difference in the second lab study.

Besides this, nearly nobody opened the "more information" box and the participants hence mostly did not see information about the encryption state of the connection for example.

5.5.8 Discussion and Limitations

Within this chapter we described an example of user intervention mechanism that uses no real detector and instead limits the number of false positives by the means looking only at cases that involve critical types of data. When starting to use such a concept a user would at first get a large number of warnings for websites that are actually no threat at all. As we showed in our field study this number would constantly drop as the user continues to make use of the method. In addition to that the users start to learn that they get positive feedback on websites they revisit and where they reuse critical data. We think that this could generate a certain learning effect for the users to understand that they should look our for possible threats whenever being on a previously unvisisted website that involves critical data. The number of false positives of our plugin that appear could be greatly reduced by using a prepopulated whitelist that would contain known trustworthy Internet parties. In such a case the user would not have to add each of his standard websites manually. On the one hand this could reduce the habituation towards our dialog and keep the user sensitive for appearing warning messages. On the other hand this could reduce the learning effect that we try to introduce using our plugin that once a website was confirmed the user will get the positive feedback because of his prior confirmation.

There are also possible technical attacks towards our concept that we did not mention earlier. In our current implementation we inject our warning window into the source code of the actual website. Attackers knowing this could mess around with the injected warning to automatically close it using their own code or by placing elements on top of our appearing dialog. If such a concept would be deployed as a real tool it would be necessary to place the warnings in a safe environment were they could not be modified or hidden by code included in the website.

Another possible issue is that although we block form submission whilst our warning dialog appears, the attacker could transmit the form input data in the background whilst the form is being filled in by the user. As a first line of defense we already use a copy of the input field in our warning dialogs. Like this input listeners to the original input field would cease to work as long as our dialog is open. In case the attacker would transmit every key press on the website he would still be able to capture the inputs into our warnings. Again, a trusted environment for our warnings would be the key to solve this issue.

Concerning our second design we introduced one major issue by hiding the important information about the encryption status of the website within a “more information” section. As we noticed most users did not unfold this section an could hence not see this information. In the first prototype this information was visible at the first glance. As an important lesson for future warning designs it is important to make such information visible immediately. Only information that is really not necessary to judge the security properties should be made accessible in a second step.

5.5.9 Research Results

Within this project we presented a research approach that introduced several new parameters to user intervention methods. A semi-blocking dialog that appears smoothly during user interaction but still blocks further interaction of the user towards the website. As a second novelty we present the dialog only in cases were critical data is involved and are hence able to reduce the amount of warnings to a minimum. This data type based property also enables us to know where the user's current focus is and we can hence introduce the warning message right at the point where the user currently interacts conveying a rational reason for the appearance of the intervention.

DD *What is Phishing Detection?*

Within this project we did not have a real detector component that processed the websites the user visits to find malicious ones by itself. Instead we limited the number of critical websites by the means of context. In our case the context of the type of input information that is involved in the online transaction. This can be seen as means of reducing the overall input space but is not a real detector as this system does not perform differently on malicious and original websites and hence does no real "detection" by itself.

DH *How can HCI be Used to Build Detectors?*

Using the users' context for filtering out inputs towards a possible detector can be seen as incorporating HCI into the detection building process. In our cases we used the reduced number of websites that we filtered using the context directly for our user intervention method. In other cases an additional detector could be used before or after the filtering process.

DR *What Detection Overhead and Thresholds are Reasonable?*

Within our field study, our participants saw a large number of 229 warning dialogs that appeared within the week. On the first day of our field study a warning appeared for 22.4% of all domains that the user visited – about one warning on every fifth website. After one week this number had been reduced to 4.5% of all the domains that were visited on that day. Our participants had not reported that any of those websites denoted a security threat and hence these numbers can be seen as false positives of our approach. A detector with such a false positive rate would be far from perfect. However, in case of this project we had the secondary goal of raising the users' sensitivity towards their private data and of making them think of were to submit this data. Together with the 229 warning we produced 552 whitelist events that were thought of encouraging people of using a website. Summing this up, in case we are able to positively reinforce the users' security behavior in a large number of cases it can also be okay to confront him with problems in the same domain as this can help to raise his overall security interest and knowledge.

IH *How can HCI be Used to Enhance User Intervention Mechanisms?*

Within this project we made use of HCI principles in a lot of different ways. The very basic concept looks at the users' behavior on the Internet and at the kind of actions a user performs online that are really security critical – in this case handling sensitive data. We then developed our concept around certain properties that would make the warning appear more sensible towards the user (see research question "IE" for more). Using a multi step approach with lab and field studies we further enhanced the warning design to fit the user needs best.

IM *How can User Intervention be Measured?*

In case of this work we did both possible types of measurements for user intervention methods. A lab study study to quantitatively assess the users' performance in detecting phishing websites was done twice to see how our user intervention methods would outperform a control condition in dangerous situations and in between a field study that was used to find out more about how people would actually use the user intervention method in their daily life and to see how our concept would evolve during a period of longer usage.

In case of the lab studies we found that the types of users that participate in such a study can make huge differences in the study outcome. In our second study the number of expert users was higher than in our first study which seemed to result in the fact that our control group did perform so well that our plugin did not make a significant difference anymore. This makes it even more important to try to counterbalance the number of security experts in such a study or even reduce it. However, this needs to be accomplished without priming the users for security. We asked them for their knowledge of "Internet technologies" in general but found out in the post questionnaire that this was different from their actual knowledge about phishing.

IE *How to Enhance User Intervention Quality?*

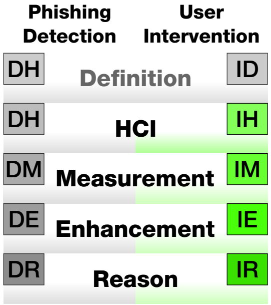
As explained in the beginning of this section we used findings from HCI to introduce several new parameters to user intervention methods in case of this work. Our "semi-blocking dialog" is a dialog that appears in context of the user's current action and is dependent of the critical data that is involved. Introducing such properties can not only be used to reduce the number of warnings but also to make sure that the warning is noticed and to raise the users willingness to pay attention. Bartsch et al. [23] did follow up work and confirm that the context can lead to a higher understanding of the risks.

IR *When Should Intervention be Performed to Which Extent?*

For this project this research question is closely coupled to the question about detector reasoning (DR). By using our context based filtering and our other new approaches for the user intervention design we were able to raise the attention to our warnings by using means that were reasonable and understandable for our participants. As discussed in the discussion section of this chapter an approach like the one presented here lies between the advantages of repeated exposure for learning purposes and the problem of warning habituation.

5.6 Enhancing SSL Awareness in Web Browsers

This chapter is based on the work that was part of the bachelor thesis "Enhancing SSL Awareness in Web Browsers" by the student Tobias Stockinger [279]. Some parts of the project also led to a publication at the 13th IFIP TC 13 International Conference (INTERACT2011) by Maurer, De Luca and Stockinger titled "Shining Chrome: Using Web Browser Personas to Enhance SSL Certificate Visualization'" [184].

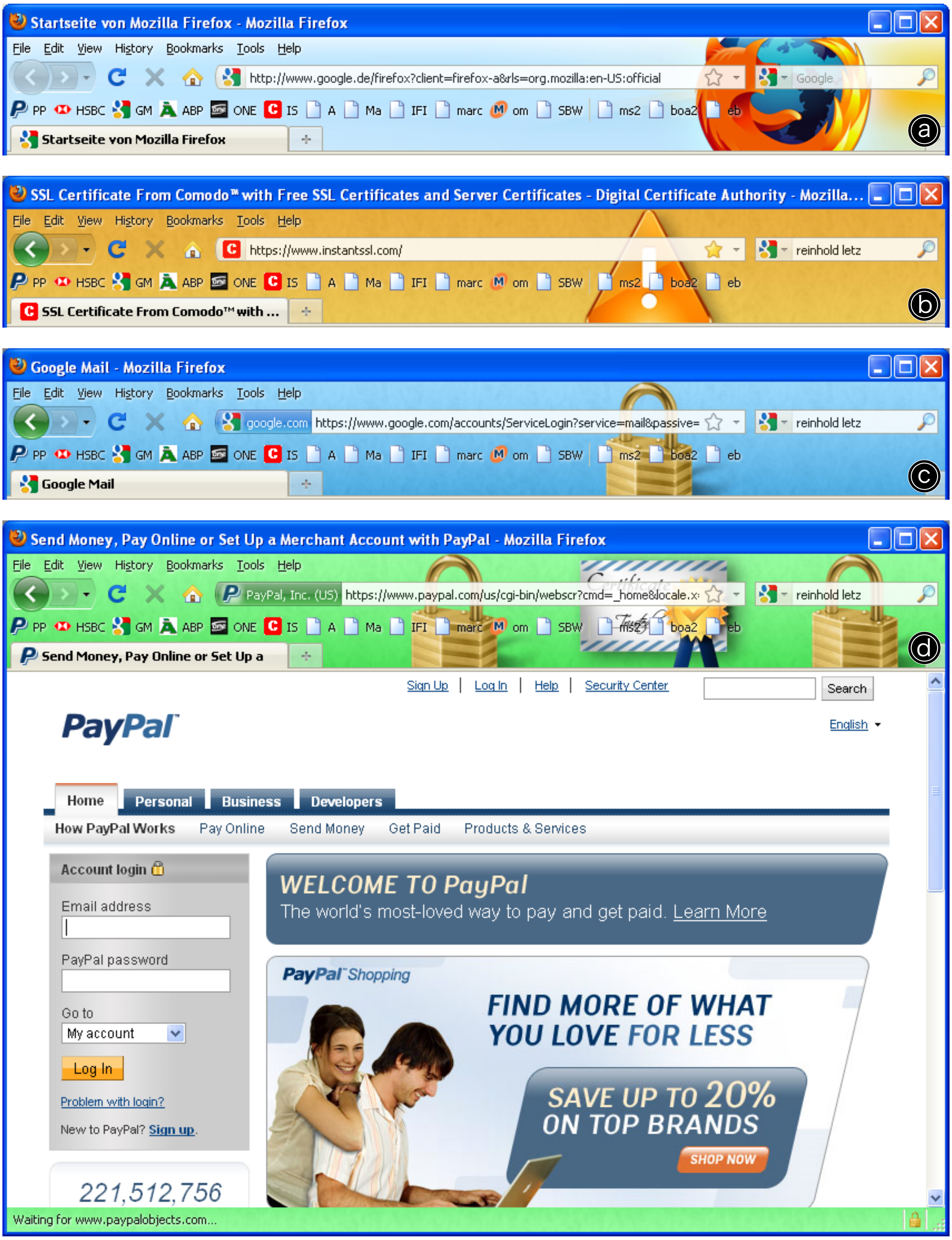


The fact that non-blocking warning messages seem to be overlooked by the participants has been often discussed within this thesis so far. For projects that have been previously presented and which used small non-blocking indicators we were able to observe a similar effect (see subchapter 5.3) but blocking user intervention methods can also lead to habituation and resentment of the users. In the last chapter we showed our idea of semi-blocking dialogs that are somewhere in-between both worlds, but within this chapter we want to take a closer look at whether it is really impossible to achieve any effects with non-blocking indicators. We look at a concept to change the whole browser interface to denote certain security states and measured whether this can have any effect on the users' assessment of security towards websites. In the SecurityGuard project in subchapter 5.2 we already used a similar approach of coloring the browser, based on our overall ranking but we did not specifically focus on this feedback. Besides that we also tested a redesigned set of the certificate warnings that appear in the browser to see whether we could generate more valid user decisions.

5.6.1 The Concept of SSLPersonas

Modern browsers, like the Mozilla Firefox browser for example, allow their users to choose skins that change the look and feel of the browser in certain areas. In the Firefox browser a lightweight skin that does only change the background images and not all UI elements is called "Persona". These Personas can be selected online and immediately be "worn" by the browser. Hence, switching between Personas can be accomplished fast (see figure 5.26a for the standard Firefox Persona).

A major advantage of these Personas is that they occupy a large amount of screen estate – and are hence well visible to the user – without using up additional space. The background area of the browser user interface so far is not used for any information display. This is


Startseite von Mozilla Firefox - Mozilla Firefox

http://www.google.de/firefox?client=firefox-a&rls=org.mozilla:en-US:official

SSL Certificate From Comodo™ with Free SSL Certificates and Server Certificates - Digital Certificate Authority - Mozilla...

https://www.instantssl.com/

Google Mail - Mozilla Firefox

https://www.google.com/accounts/ServiceLogin?service=mail&passive=

Send Money, Pay Online or Set Up a Merchant Account with PayPal - Mozilla Firefox

https://www.paypal.com/us/cgi-bin/webscr?cmd=_home&locale.x=

WELCOME TO PayPal

The world's most-loved way to pay and get paid. Learn More

FIND MORE OF WHAT YOU LOVE FOR LESS

SAVE UP TO 20% ON TOP BRANDS

Waiting for www.paypalobjects.com...

Figure 5.26: A Firefox web browser "wearing" a standard Persona skin (a) and our own Personas for the different states: b) Warning (partially unencrypted content) c) Standard SSL certificate d) Extended Validation SSL certificate (based on [184]).

why we wanted to use these Personas as means to visualize the current encryption state of a website the user is currently visiting.

Other indicators like the lock icon or even the colored indicator in front of the URL bar are usually overlooked by the users as confirmed by related work. The effect of "change blindness" [267] most certainly is part of the explanation for this issue found in many experiments. Rensink et al. [245] found that especially in marginal interest areas of an image, large changes may go completely unnoticed by the viewer for a long period of time. Increasing the size and kind of the visual change using Personas we want to counteract the effect.

Besides using an unencrypted connection there are three possible states an encrypted connection can have. In case a website is SSL protected but some of the website content is loaded using unencrypted connections this is stated as being problematic, as content that is transmitted over those channels may be subject to eavesdropping. In case a connection is properly encrypted, two types of SSL certificates exist. Standard SSL certificates that are issued by certificate authorities and extended validation SSL certificates that require an additional validation of the company to include company details within the certificate.

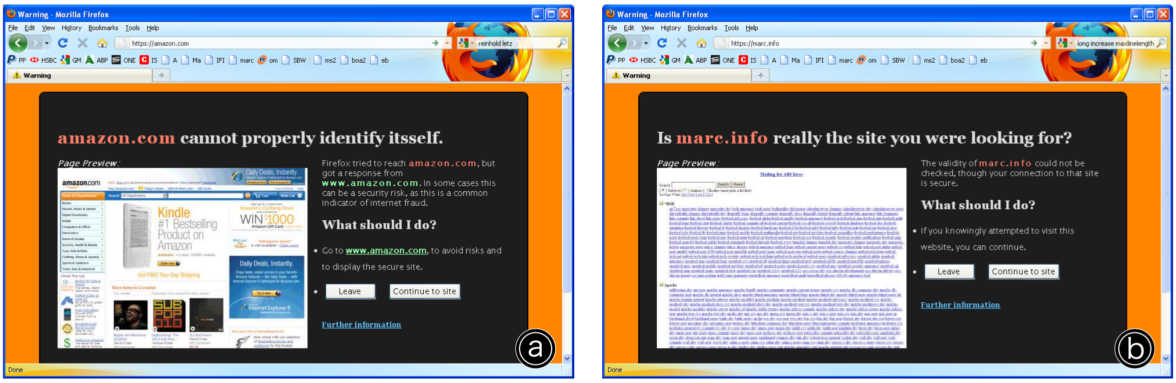
For those three cases we designed three different kinds of Personas that can be seen in figure 5.26. A warning Persona colored the browser yellowish with an additional warning sign containing an exclamation mark in the background. The Persona for standard SSL was colored blueish containing a huge lock icon and the extended validation certificate was colored greenish containing two lock icons and an additional certification icon. The basic color scheme used by us is identical to the colors used by the Firefox browser itself for the different states.

5.6.2 Redesigning SSL Warning Messages

In all of the above cases the certificate used could be verified by the browser and the website would be displayed. However, a few other cases of encrypted websites exist were the certificate cannot be verified by the certificate authority and the browser hence displays a warning before letting a user access this website. These warnings usually appear if a certificate has not been signed by a certificate authority known by the browser (self-signed certificate) or if the "common name" – the stored domain name – of a certificate does not match with the actual URL the certificate is applied to.

In both cases Firefox displays a warning screen before giving the user access to the respective website. As a lot of university or company specific websites use self-signed certificates to save the money needed for certification, many users simply learned how to skip the warning message although they do not understand its contents.

For both of the aforementioned cases we redesigned the warning messages and tried to make them more usable. On the one hand we included a huge preview image of the respective website to make each warning look more unique and to give the user a preview of what the


Warning - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://amazon.com
reinhold letz
PP HSBC GM ABP ONE IS A Ma IFI marc om SBW ms2 boa2 eb
Warning
Page Preview:
Leave
Continue to site
Further information
a
Done
Warning - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://marc.info
long increase maxlinelength
PP HSBC GM ABP ONE IS A Ma IFI marc om SBW ms2 boa2 eb
Warning
Page Preview:
Leave
Continue to site
Further information
b
Done

amazon.com cannot properly identify itsself.

Firefox tried to reach amazon.com, but got a response from www.amazon.com. In some cases this can be a security risk, as this is a common indicator of internet fraud.

What should I do?

- Go to www.amazon.com, to avoid risks and to display the secure site.

Is marc.info really the site you were looking for?

The validity of marc.info could not be checked, though your connection to that site is secure.

What should I do?

- If you knowingly attempted to visit this website, you can continue.

Figure 5.27: Our redesigned SSL warning messages containing thumbnails: a) Mismatched common name (domain) of the certificate, b) self-signed certificate error (based on [184]).

site that she wants to access would look like. These preview images were loaded from an online service without really loading the website. In addition we recreated the wording and different options of the warning dialogs to highlight important terms and make them more understandable. A "further information" link was provided to users for even more details on the problem. When designing those warnings we stuck to a number of previously published recommendations by other researchers (e.g. [78] and results found in previous projects). We took special care of avoiding technical terms and lengthy messages whilst providing clear choices and preventing habituation. Figure 5.27 shows the two redesigned warnings in action.

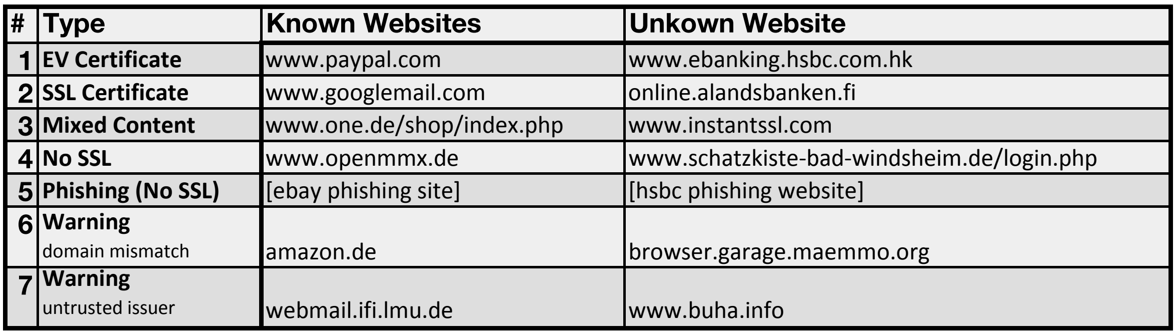
5.6.3 Lab Evaluation

For the lab evaluation of our concept we wanted to find out whether the Personas displayed by our plugin would be able to influence the users' opinion for a specific website towards more positive or more negative ratings.

Methodology

In a mixed-model design design we assigned 24 participants to between-subjects groups using our plugin or using a standard browser (12 participants in each condition) and let them browse 14 different websites of seven different categories. After viewing each website we asked the participants to rate security and trustworthiness – our dependent variables – of the given website. Up to the end of the study we did not give any explanations about the plugin or the meaning of the different Personas.

The 14 different websites came from two other within-subjects variables that we had. On the one hand we used websites from seven different categories to test all aspects of our plugin: 1) websites having a proper extended validation certificate; 2) websites using a standard



#	Type	Known Websites	Unkown Website
1	EV Certificate	www.paypal.com	www.ebanking.hsbc.com.hk
2	SSL Certificate	www.googlemail.com	online.alandsbanken.fi
3	Mixed Content	www.one.de/shop/index.php	www.instantssl.com
4	No SSL	www.openmmx.de	www.schatzkiste-bad-windsheim.de/login.php
5	Phishing (No SSL)	[ebay phishing site]	[hsbc phishing website]
6	Warning domain mismatch	amazon.de	browser.garage.maemmo.org
7	Warning untrusted issuer	webmail.ifi.lmu.de	www.buha.info

Table 5.11: List of the 14 URLs used for the SSLPersona lab study. For each of the seven categories of websites we had one known and unknown website per category.

SSL certificate; 3) websites that had mixed content within the encrypted website; 4) genuine websites that did not use SSL; 5) phishing websites that did not use SSL and; finally two conditions that triggered the display of our new warning dialogs. Firstly the mismatched domain warning (6) and secondly the unknown issuer (self-signed) warning (7). Using the Alexa ratings of different websites we used one well-known and one less-known website for each of our conditions thus ending up with two within-subject variables and 14 different tasks per user (see table 5.11 for a complete list of URLs).

We showed the 14 websites in random order to our participants to avoid ordering effects. After viewing each of the websites the participants had to answer a set of questions. For the websites regarding the Personas (1-5) we asked four questions about them knowing the website in general, a trustworthiness rating from -2-“this website seems suspicious” to $+2$-“this website seems trustworthy”, a security confidence rating from -2-“I cannot see whether this site is secure” to $+2$-“there are enough indicators that this site is secure/insecure” and an assessment about the willingness to login to that website.

Concerning the warning messages we asked 5 different questions on Likert scales from -2 to $+2$ covering the following areas: understanding of the warning message content; ease of understanding the warning message content; did they read the entire text; perceived length of the text and an assessment of the severity of the warning.

Results

The age of our 12 participants in the control group (standard browser with no plugin) ranged from 14 to 45 years (average 27) with 8 male participants. In the plugin group we had an average age of 23 years (14 to 30) and 7 male participants. Our participants used the Internet on average for several hours each day and 20 of the 24 participants did online banking.

We first look at how well our classification of known and unknown websites had worked. In case of the conditions 1,2 and 5 our classification was correct but for the other two conditions the selected well known websites were actually not known very well. However, the unknown websites were truly unknown for all our conditions and as the overall results of both levels were very similar we will only report the findings from the seven unknown websites here.

	SSL							No SSL			
	Extended Validation ①		Standard SSL ②		Partially not encrypted ③			Genuine ④		Phishing ⑤	
Trustworthiness											
Median	1,5	1	1	0	-1	0		0	0	-1	-1
Mean	*1,50	0,83	*0,67	0,58	*-0,50	0,17		*-0,33	0,17	*-0,58	-0,5
SD	0,52	0,94	0,89	1,00	1,24	1,11		0,89	0,72	1,56	1,68
Would login											
Median	1	1	-0,5	-0,5	-1	0		0	-0,5	-1,5	-2
Mean	*0,92	0,33	* -0,5	-0,6	* -0,5	-0,2		-0,4	* -0,5	-0,8	* -0,8
SD	1,16	1,23	1,38	1,44	1,38	0,94		1	1,17	1,4	1,53
Can determine security											
Median	1	0	-0,5	-1	0	-1		-1	-0,5	-1,5	1
Mean	*0,58	-0,1	* -0,1	-0,7	* 0	-0,7		* -0,9	-0,7	-0,8	*0,42
SD	1,51	1,08	1,24	1,44	1,21	1,23		1,08	1,3	1,47	1,38

Table 5.12: The rating results (medians, means and standard deviation) for the seven unknown websites in the five conditions belonging to the Personas. A star indicates for each condition and question whether our plugin had the intended influence or not [184].

We found a slight tendency that our plugin had a larger effect on people if the website was unknown.

Looking at the average answers of our participants towards the three remaining questions in the different conditions one can see from table 5.12 that the plugin influenced the users' behavior as intended for most of the conditions – especially for the Persona based conditions 1 to 3. In case of the conditions 1 and 2 we expected that the positive indicators in our plugin group would raise the users trustworthiness towards the website, their willingness to login and their perceived ability to determine security. This was the case for all the answers in this category. In case of the partially not encrypted website we expected that trustworthiness and the willingness to login would go down (which they did) but that the ability to determine security would still be rated higher – this was also the case.

In case of our non-SSL secured websites (conditions 4 and 5) our plugin did not play any direct role, as it did not display a Persona for any user group. What we wanted to look at for these two conditions were carryover or learning effects from our plugin. A missing feedback of our plugin finally should result in an increased suspiciousness of the users. In fact we were not able to observe such an effect within this lab study. Due to the short duration of our study, familiarization with our plugin could not happen. Besides, showing the websites in random order to our participants proved to be a methodological flaw here because it meant that some participants of the plugin group saw those conditions without ever having been exposed to the Personas yet.

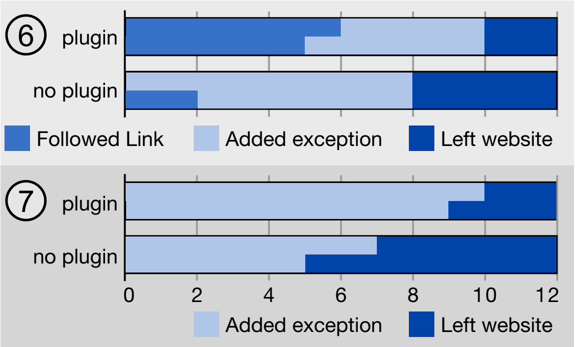


Figure 5.28: Decisions taken by the participants for the mismatched domain warning (condition 6) and the self-signed certificate error (7). The upper part of a bar represents the decisions for the well known websites (based on [184]).

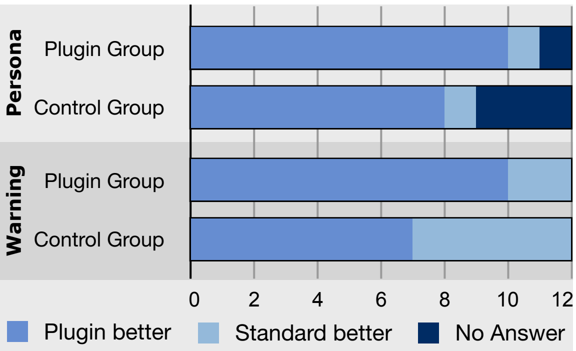


Figure 5.29: Acceptance of the SSLPersonas concept and the newly designed warnings throughout the participants (based on [184]).

In case of our redesigned warning messages the behavior of our plugin-group participants was much more rational than the behavior in the control group. In case of the mismatched domain warning it will usually be best to visit the website that the certificate was originally intended for. For the unknown websites this was done by two people in the control group and five people in the plugin group. In case of the known websites the effect was even stronger (see figure 5.28). Condition 7 was dedicated towards self-signed SSL certificates (untrusted issuer) and people could only leave the site or set up an exception. As we only had genuine websites in our study that used self-signed certificates it was okay to set up an exception which was done more often by our plugin users. Comparing those results statistically using a repeated-measures ANOVA the differences for the between-subjects variable “plugin” are highly significant ($F_{1,22} = 16, p = .001$).

In the end we debriefed all our participants and explained them what the study was about and how our plugin worked. We showed all participants of both groups side-by-side images of our concept and wanted them to vote for a preferred version in terms of the Personas that they saw and concerning the new warning messages. A majority of both groups voted in favor of our Persona (75%) and warning (71%) concept whilst the participants that had actually experienced the plugin were even more satisfied (see figure 5.29 for details).

5.6.4 Field Evaluation

In August 2010 we published the developed plugin and deployed it on the official addon website of the Firefox browser. It was downloaded several thousand times and security related blogs and podcasts featured it (e.g. [106, 209]). By March 2011, when the corresponding paper was written we had more than 15,000 downloads.

Methodology

We wanted to use this real world user base to find out about the qualitative performance of our plugin and compiled a questionnaire that was available in German and English (the languages our plugin was most often used in). Besides demographic questions the survey contained several questions from the "IBM Post-Study System Usability Questionnaire" [161] using a 5-point instead of the original 7-point Likert scale and finally some questions about the plugin usage and security knowledge of the participants. We included a popup with an invitation to the survey within the next update of our plugin.

Results

Within a period of two weeks 169 users of our plugin had filled in our questionnaire. The age of our users ranged from 9 to 70 years (41 years in average) and 9.5% of the participants were female. They used the Internet in average 30 hours (SD 23.1) per week and 28% stated to have expert computer skills. 9% stated that they had been attacked by a phishing attack before but only one single participant was successfully attacked.

Since we had no technical means of measuring how long the plugin had been installed on the users' computers we asked the participants in the questionnaire. 15% used it less than one month, 40% 1-2 months, 30% 3-5 months and 15% more than five month. Between those groups we did not see any real differences in their answers besides that it seemed that the number of experts was higher in the group of long-term users. We think that it could be that the plugin had firstly been discovered by more experienced users and that the novice users started to adopt it later.

We asked some questions about the existing Firefox SSL indicators (in front of the URL) and only 82% stated that they had seen this indicator before. Interestingly, only 11% were able to give a correct explanation. 50% at least stated that they had clicked the area before.

Concerning our plugin we wanted to see whether the users understood the different Personas. 62% stated that they knew the difference between our green and blue Persona but only 21% were able to give a correct textual answer. Another 21% mentioned the "strength of encryption" in any way which is completely wrong as both certificates can be used with the exact same encryption methods and key lengths. This shows that although many users had classified themselves as being experts they were not able to correctly explain the SSL related concepts. 38% of the users stated that SSLPersonas had changed the way they use the Internet. Concerning the IBM Usability questionnaire the users' answers were mostly positive (see figure 5.30).

5.6.5 Discussion, Limitations and Future Enhancements

Within this project we presented a user intervention mechanism targeting SSL warning dialogs and SSL visualization within the browser. As the SSL status can only identify whether

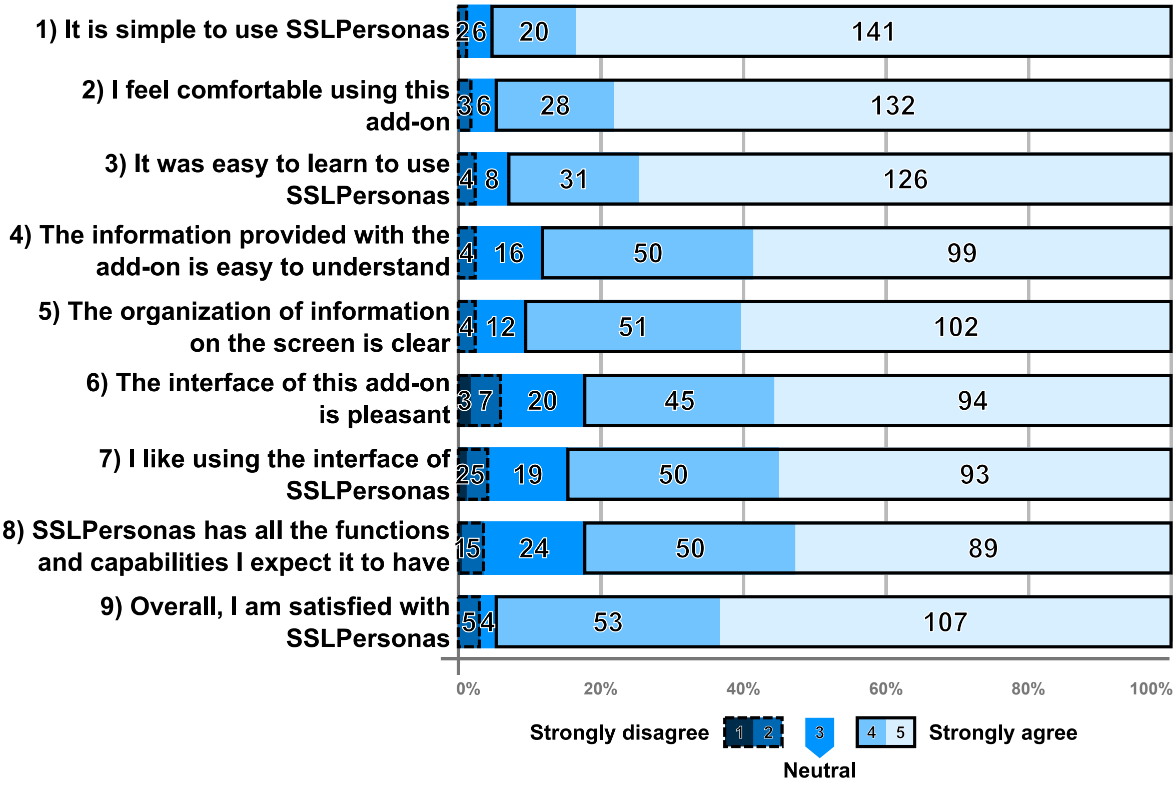


Figure 5.30: The users' answers concerning the SSLPersonas nine usability questionnaire questions (based on [184]).

the connection itself is protected from eavesdropping it is not guaranteed that the party sitting at the other end of the connection is not a malicious one. Phishers might as well use SSL encryption for their phishing websites although they usually do not – as these certificates cost money and they have to acquire the signed certificates somehow.

With our concept of SSL personas we showed in our lab study that we are able to influence the users way of thinking about encrypted websites in a positive way. As some of the participants of our field study confirmed it may change the way one uses the Internet. Whether or not our plugin leads to any long term effects on websites that are not SSL encrypted could not be shown within our lab study and was also not part of the field study.

Using the questionnaire of the field study we found out that a lot of users (long term and short term) classify themselves as expert users. This seems interesting as our plugin is mostly intended for novice users. We suppose that a huge problem for novice users is to find such enhancements like plugins by themselves. They either do not actively look for browser extensions like ours and if so, they are perhaps unable to install those. We hence recommend that novice security features should be shipped as standard features of a software and for experts the possibility of disabling such features should exist.

Another issue that was often mentioned when discussing our field study was the fact that the high rating of our participants might simply arise from the fact that they are long term users of our plugin and would not use it, in case they did not like it. Nevertheless we think the way of evaluating software by deploying it into the real world and collect feedback from real users might be much better than artificially force a few participants to use a piece of software. Biasing may occur in both ways as participants might try to please the experimenters with their ratings and an active user base for any software can already be seen as some kind of quality metric. In the research domain around mobile devices such evaluation methods are already done by researchers like Henze et al. [122] for example.

When evaluating the questionnaire data from our plugin we looked closer into the fact that the same amount of users that correctly explained the difference between our green and blue Persona stated that the green Persona would denote stronger encryption. We found that within our green Persona image we used two lock icons compared to one lock icon within the blue Persona. This might have misleaded the users into thinking of a stronger encryption. In a later version of our plugin we replaced the second lock icon with the icon of a green man to denote the certified identity.

Looking at the new warning designs that we proposed we managed to get significantly more people into choosing correct decisions within such a dialog. We attribute this to our design types that showed clear choices with less text than the original warnings and created more individual looking dialogs using formatting and imagery.

5.6.6 Research Results

As this project provided only SSL security indicators of different ways we had absolute no kind of detection process involved. But although our Persona feedback components were passive and non-blocking we were able to prove that these non-blocking indicators can have some influence on the user's opinion towards a website.

IH *How can HCI be Used to Enhance User Intervention Mechanisms?*

Non-blocking security indicators are mostly overlooked as the users focus when visiting a website is skewed towards the website content and security is never a primary goal. This means that a non-blocking security indicator has to fight for attention, but screen real estate is precious to the user and nobody wants to have 30 percent of his screen reserved for security feedback (see the project in subchapter 5.2). Using the Persona concept we were able to use a relatively large portion of the screen for security feedback without having to reserve additional screen real estate. The research about "change blindness" seems to be important related work when it comes to non-blocking indicators. Besides the property of more space other properties might also be used to achieve a higher amount of attention. these measures are always coupled with the research questions around "reasoning" as too much of an alarm for any tiny security issue, will set up the users sooner or later.

Concerning our newly designed warning messages the guidelines that have been proposed by other researchers combined with our own findings so far provide a good basis for designing warning messages that might finally lead to more correct decisions.

IM *How can User Intervention be Measured?*

As in the project in subchapter 5.3 it is not always necessary or possible to measure the achieved correctness of user behavior. In case of SSL certificates it would make no practical sense of testing how users would behave on SSL secured phishing websites as they hardly exist. Instead of measuring achieved correctness, an encrypted website should make a user more confident in using it and this can be measured by self reporting as it was used within this project. Using such a method, a baseline to compare against is very important, as the measured values depend to a large extent on the website contents and other factors. Only the difference in both measurements (with plugin/without plugin) can yield valuable insights.

For our warning messages we were again able to evaluate the user behavior to see whether it would lead users to more correct decisions.

IE *How to Enhance User Intervention Quality?*

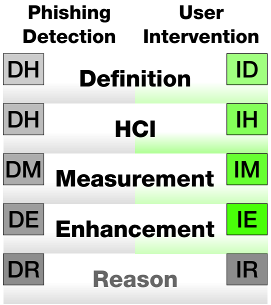
First of all this chapter showed that non-blocking indicators are not per se useless for conveying warning information. It is possible to providing indicators that are noticed by the user although security is not the user's primary by getting various properties right. Overall a certain visibility or notability can only be achieved for indicators having a certain size and stand out against the rest of the user interface. We used large background graphics of an application in this case. When designing such elements the details that are communicated by the imagery and content have to be reviewed. In our case using two lock icons in one image whilst using only one lock icon in another graphic most probably made the users think of one type of certificate denoting a stronger encryption than the other.

IR *When Should Intervention be Performed to Which Extent?*

Reasoning about the amount of user intervention is especially important within this project as the intervention does appear in many situations and mostly in non-critical instead of critical ones. As we saw in the previous chapter such positive reinforcement denotes another possibility of helping the user with security. In contrast to messages that only warn about threats positive security reinforcements can have the effect of creating a positive attitude of the user towards security principles and may even lead to some kind of learning effects. However, positive indicators that are not reporting any kind of threat are generally expendable and hence should never be too flashy. This is why the concept of a non-blocking indicator, as it was used within this project, is a very good example for such a user intervention method.

5.7 Diminishing Visual Brand Trust

This chapter is based on the work that was part of the bachelor thesis "Diminishing Visual Brand Trust on Websites for better Security Assessment" by the student Cornelia Reithmeier [244].



So far we saw in the previous subchapters that it is possible to influence the users' security opinion towards a website to some degree even when using non-blocking indicators. However, the part of a website that still catches the largest amount of attention is the content area of the browser. From the point of view of usable security this is at the heart of the problem because users are blinded by the look of the content that can be so easily impersonated by an attacker. Within this subchapter we want to present a project in which we tried to have a look at whether it would be possible to get that strong content focus of a user lowered by making the content appear less trustworthy. We built a browser plugin that exchanged logo images of a website and replaced them by images of other websites or modified the text appearance of a website. After some initial research we conducted a focus group to find out what others thought about our concept before finally developing a plugin that took care of the content changes. Finally, we conducted a user study to see whether our modifications would help to change the users' security behavior.

5.7.1 The Concept of Destroying Content Trust

When browsing the Internet the users' primary focus is never on security and most of the time the user is focused on the visual content area of a web browser where the actual website is displayed. Due to this, mobile browsers for example are completely hiding the browser interface whenever possible. For the concept of this project we asked ourselves how we could make users more aware of the other indicators in the surrounding browser chrome. On the one hand this can be achieved by the flashiness of the placed indicators around the content (see subchapter 5.6) or by blocking the access to the chrome content (cf. related work or the project in subchapter 5.5). But is there a way to make users focus other security indicating elements by breaking the users' trust into the content area to some extent?

Using existing related work and some own research we compiled a list of elements within a browser that can be seen as generating trust for the users' actions [**?**, 95, 275]. We use the term "trust" instead of security within this subchapter because the users' motivation to continuously carry out actions on a website is more than just security. The different aspects of the user interface all add to the fact that users keep executing their online transactions. If they would not "trust" the sum of the whole interface they would suspend those actions. The following list of items hence adds to this trustful user experience in some way:

- **Website Design:** Although a very broad term, the design of the whole website a user is visiting, is what defines the brand of a website and helps the users to recognize known companies they are usually doing business with.

- **Website Logo:** As a subelement of the design, the logo of a company is the most unique part for brand recognition. A big problem about logos in the digital world is the ease with which a logo can be copied. Even in the physical world the fashion business and other industries have to fight against logo misuse although faking a logo physically is much harder then just copying an image file from one server to another. Logos as a key element of impersonation have been subject to other research specifically looking for logo impersonation [254].
- **URL:** The URL of a website really tells the users which Internet server they are currently communicating with. Compared to a logo in the content area it is much harder to fake this information.
- **Encryption Status/SSL:** This tells the user something about how the data between both communicating parties of a web server connection is handled and whether it is protected from eavesdropping.
- **Certification Seals:** Coming from the physical world it has often been practice that some governmental or private companies are used to certify others and their services. In Germany a famous example is the "TÜV" [285] (short for technical surveillance association) that has the duty of checking every car every two years for its ability to participate in road traffic. But these kinds of certifications are not limited to the offline world. The TÜV and other companies also certify online retailers and these are then allowed to place a respective seal on their website. As with the logos such imagery is easily copied.
- **Links:** Linking to other websites or to websites within the own website makes a website appear connected with other parties of the web. Hence, they allow the user to assess the relationship of the current website to other websites to some extent. It is important to note that these certification seals and their images have nothing to do with encryption certificates.
- **Third Party Logos:** Logos of other companies may also generate trust. If one company does business with another company there must be some trustful relationship between those.
- **Personalization:** Most Internet companies state that the users should look for personalization within email or other communication they receive to validate the other company. Often attackers are only in possession of the email address of a user and would not be able to generate personalized content.
- **Security Instruction:** Security instructions on a website that tell the user for example how she is able to verify that she is communicating with the right party may be helpful to some extent but they can also backfire [136].

Taking these items into account we made a first list of website properties that could be altered to hopefully change the users reliance from content related indicators to other ones that are more reliable from a security point of view. Such a concept would be very invasive towards the users' web browsing actions which is why we postponed a decision whether or not the concept would be suitable for a deployed browser solution to after our focus group (see section 5.7.2). Nevertheless a general investigation of a possible change in user behavior would also be interesting.

- **Replacing Logos:** Logos on a website could be replaced by other logos or could be left out.
- **Replacing Images:** The same is true for other imagery of the website. Despite that other pictures or image-based design elements might be less known than a logo, a replacement with other images could still change the look of a website.
- **Changing Image Color:** Changing the image color especially of a logo or a photo would distort the natural look of an image without completely replacing it. A problem here some companies really use their logo in multi-colored versions.
- **Changing Website Color or Style:** Besides images a website contains other style elements like the layout, layout elements or text colors. All these attributes of a website can be easily changed to end up with a totally different looking design.
- **Changing the Position of Interface Elements:** Each website uses a certain kind of navigational structure or other kinds of UI elements. An example for distorting the user interface in this way would be to move a menu bar from the left of the website to the right.

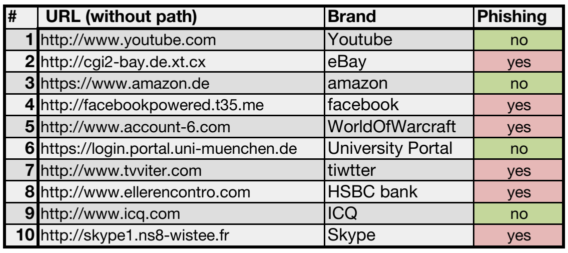
5.7.2 Focus Group

After we had identified possible properties of modifications we wanted to discuss the whole concept within a focus group. Our goal was to find out more about what properties of websites make a website trustworthy and how those would be ranked by our participants. As a last part of the focus group we wanted to gather insights about how others think about our modification and study ideas.

We had six participants joining our focus group (one female) with an average age of 23 years. Four of our participants were studying informatics or media informatics.

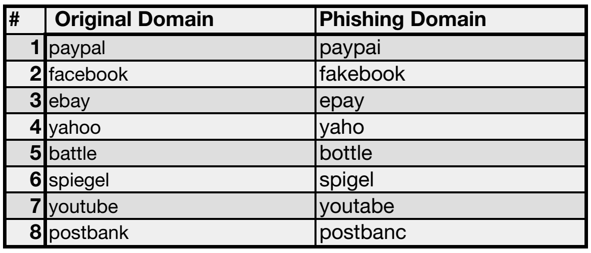
Motivation

We didn't mention the exact topic of our focus group beforehand and instead told the participants that the focus group would be about "visualization of websites". We did this to make our participants in a first phase more aware of the actual problem of people focusing more on the content of websites instead of the security indicators. In the beginning of the focus group we showed the participants 10 different screenshots of websites projected to the



#	URL (without path)	Brand	Phishing
1	http://www.youtube.com	Youtube	no
2	http://cgi2-bay.de.xt.cx	eBay	yes
3	https://www.amazon.de	amazon	no
4	http://facebookpowered.t35.me	facebook	yes
5	http://www.account-6.com	WorldOfWarcraft	yes
6	https://login.portal.uni-muenchen.de	University Portal	no
7	http://www.tvviter.com	tiwtter	yes
8	http://www.ellerencontro.com	HSBC bank	yes
9	http://www.icq.com	ICQ	no
10	http://skype1.ns8-wistee.fr	Skype	yes

Table 5.13: A list of the URLs belonging to the screenshots that we showed at the beginning of the focus group to motivate towards the problem.



#	Original Domain	Phishing Domain
1	paypal	paypai
2	facebook	fakebook
3	ebay	epay
4	yahoo	yaho
5	battle	bottle
6	spiegel	spigel
7	youtube	youtabe
8	postbank	postbanc

Table 5.14: A list of the eight different domains used for the study and the phishing domains that were used to replace the respective domains depending on the task.

wall of the conference room that was used for the focus group. Six of those websites were actual phishing websites which could easily be seen when for example looking at the URL. We handed a list to each participant and asked them to fill in the name of each website and special elements that attracted their attention when they saw the website. For looking at the screenshots and noting their answers they only had fifteen seconds. The results of this first motivational task were later on not considered for any analysis. Although a lot of the URLs we had chosen were easily recognizable as being phishing none of our participants had noted that they saw some phishing websites among the original ones. A whole list of the URLs used can be found in table 5.13.

Indicator Trust and Phishing Experience

After having debriefed our participants and having explained the problem once again, we started with our discussion phase. We presented the participants with the trust elements we had identified earlier, written on cardboard and wanted them to discuss the elements and finally sort them according to their importance. After a short discussion the participants had identified the five most important elements and started to rank them. They agreed that some elements are of nearly equal importance and hence ranked the logos and the overall design of a website as most important followed by the URL and SSL/https indicators and as a third level of importance they chose the certification seals.

Afterwards we asked them to tell us more about their personal experiences with phishing: whether they noticed any attacks on themselves before and if so how they had detected those. Two group members remembered that they had received phishing emails before but they said that they had not fallen for the attacks.

Introducing our Concept

In the last stage of the focus group we gave a short introduction into the idea of our concept and showed the participants some mockup images of how the modifications might look like. Examples for those mockups can be found in figure 5.31. We asked them for their opinion

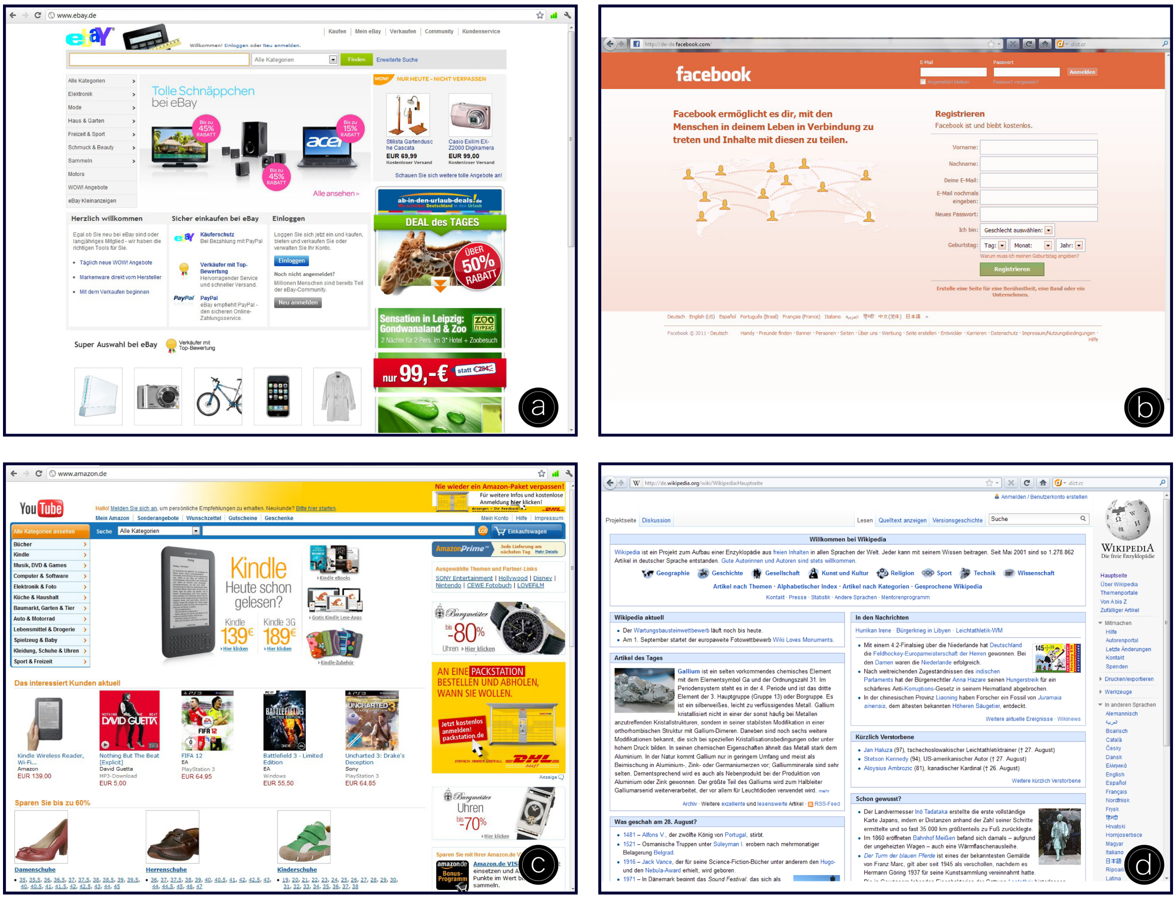
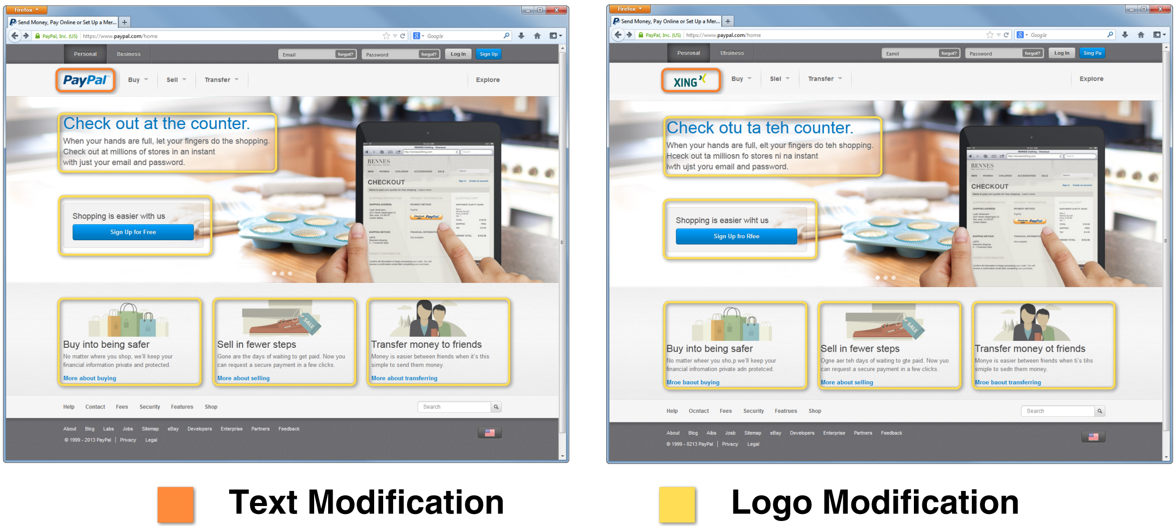


Figure 5.31: Different mockup examples of modifications that were shown to the participants of the focus group: a) Logo with changed colors; b) Changed color style of the website; c) replaced logo image; d) switched menu location.

towards the concept and its possible changes and whether they had any concerns about the idea.

The replaced logo was experienced most striking by the participants but they also had their doubts about it. They mentioned the Google Search engine as an example where the logo of the search engine is changed every few days for special occasions. Changing the color scheme of a website was commented to be possibly hard to notice if the participants are not extremely familiar with a website. Next, the participants expressed their doubts about changing the page structure (e.g. the menu position). First of all users could explain this away as being a formatting error and as a second reason this could sometimes be thought of as being simply for another cultural group.

The most interesting finding from our focus group was a completely new kind idea of modification that came up during the discussion. One participant mentioned that erroneous encodings for special characters of the German language are always a warning sign for him when he gets emails (e.g. an ‘ä’ just appears as ‘?’). Other spelling or grammar mistakes



Firefox
Send Money, Pay Online or Set Up a Mer...
PayPal, Inc. (US) https://www.paypal.com/home
Google
Personal
Business
Email forgot?
Password forgot?
Log In
Sign Up
PayPal
Buy
Sell
Transfer
Explore
Sign Up for Free
RENNES
CHECKOUT
SALE
Help Contact Fees Security Features Shop
Search
About Blog Labs Jobs Sitemap eBay Developers Enterprise Partners Feedback
© 1999 - 2013 PayPal | Privacy Legal

Check out at the counter.
When your hands are full, let your fingers do the shopping.
Check out at millions of stores in an instant
with just your email and password.

Shopping is easier with us

Buy into being safer
No matter where you shop, we'll keep your financial information private and protected.
More about buying

Sell in fewer steps
Gone are the days of waiting to get paid. Now you can request a secure payment in a few clicks.
More about selling

Transfer money to friends
Money is easier between friends when it's this simple to send them money.
More about transferring

Firefox
Send Money, Pay Online or Set Up a Mer...
PayPal, Inc. (US) https://www.paypal.com/home
Google
Pesronal
Ubsiness
Eamil forgot?
Password forgot?
Log In
Sing Pu
XING
Buy
Slel
Transfer
Explore
Sign Up fro Rfee
RENNES
CHECKOUT
SALE
Help Ocntact Fees Security Featrues Shop
Search
About Blog Albs Josb Sitemap eBay Developers Enterprise Partners Feedback
© 1999 - 0213 PayPal | Privacy Legal

Check otu ta teh counter.
When your hands are full, elt your fingers do teh shopping.
Hceck out ta milliosn fo stores ni na instant
iwth ujst yoru email and password.

Shopping is easier wiht us

Buy into being safer
No matter wheer you sho,p we'll keep your financial infromation private adn protetced.
Mroe baout buying

Sell in fewer steps
Ogne aer teh days of waiting to gte paid. Now yuo can request a secure payment in a few clicks.
More about selling

Transfer money ot friends
Monye is easier between friends when ti's tihs smiple to sedn them money.
Mroe baout transferring

Figure 5.32: Example of the PayPal home page without modifications and after our modifications (logo+text) have been applied.

where also mentioned to cause distrust. For our plugin the participants suggested to create those errors on purpose by translating a piece of text to another language and back using some automatic online translator tool. This should create similar looking errors.

Participants were also worried about the fact that our modifications could appear too often. They proposed to include a whitelist within the plugin that would keep already known websites free from modifications or that the modifications would only appear on websites where one had to enter login credentials. In general the participants did not really like the idea of having their websites modified during their daily browsing sessions as we had considered it so far. This showed us that our plugin would not be really suitable as a tool that could be deployed to end users but we at least wanted to use it to gain our research findings we mentioned in the beginning of this subchapter.

5.7.3 The Final Plugin

Within our focus group we had identified that logo and text changes seemed to be the most suitable ones. Figure 5.32 shows an example of the final algorithms applied to the standard PayPal home page.

For the final plugin we decided to use logo replacement instead of logo recoloring. For that the current logo of a website should be replaced by a randomly chosen logo of a different brand. We discussed several possibilities of how to create the logo repository to choose from. The solution would have been to extract the logos from other browsing sessions of the same user to only swap the current logos with other logos that are known to the participant. A problem with this approach was that it might lead to privacy issues disclosing the brands that a user usually visits. Because of this issue we started with a fixed set of well known logos in the German language area. When replacing a logo we resized it in a way that the aspect ratio of each logo was kept but placed it within the boundaries of the original logo.

The technical discovery of the logo images can get very hard as logos are not easily distinguishable from other images within a website. For our prototype we developed an algorithm that searches the image source or “id” attribute within the HTML code for an occurrence of the word “logo”. If no logo is found within the image text we then move on to anchor elements and check their attributes and contents and afterwards we check other elements and their attributes. On most standard websites this algorithm appears to be quite stable and worked well enough for our testing. In a professional environment more advanced solutions like the one presented by Jiao et al. [142] could be used.

For our text replacement we had problems of finding a free and fast online translating tool that would offer an API to quickly do the translation requests that we needed. As it was important that the text replacement would be visible at first sight of a website we could not afford to wait for long lasting online translation. In addition for each website a lot of different text areas exist that would each need to be translated twice (forward to another language and then backward to the original language) which would cause a lot of server requests. We chose to implement an own algorithm for error creation by interchanging single characters within the words of a text. Using such a method the modified text can still be understood but looks somehow weird to the user. The technical implementations of the textual modifications were pretty simple and straight forward. For each word of each text area of a website we randomly decide whether it should be changed. In case a word should be changed only one random character of the word switches its place with the next following character. The character case is preserved using the case of the original character position.

5.7.4 User Study Evaluation

With the so far developed prototype we now conducted a user study to find out whether the modifications introduced would lead to more people investigating other security indicators outside of the content area. To measure whether a participant looked at the different indicators we used questionnaire data, a think-aloud protocol and an eye tracker to see where exactly our participants looked at.

Methodology

We conducted our study as a within-subject design and used two independent variables. The first variable “modification” had four different levels: no modification, logo change, text change and text+logo change. Besides that we wanted to apply our concept on phishing and non-phishing websites. This results in eight different conditions each user had to go through. We balanced those conditions using an 8 by 8 Latin square .

Whenever we wanted to show a phishing website we used a simple browser modification that will be explained in more detail in the project in subchapter 5.9 to make the original website look like a phishing one. In case the original website was encrypted we modified the security indicators to appear as if the website was not encrypted and besides this we exchanged the URLs in the URL bar and the status bar of the browser with new domain names. We selected a list of eight different original websites that were either among the most phished websites on phishtank.com, or the most phished websites reported according to the anti-virus company Avira [211]. For each of those websites we derived a possible phishing URL by creating a similar sounding domain name that could have been registered by a phisher. Table 5.14 contains a list of the eight used domains and the respective phishing domains we used. For all our eight conditions we assigned the websites that should be used randomly in such a way that each brand was used once throughout one user trial.

Using the eye tracker, think-aloud, the questionnaire and additional logging we collected different dependent variables from three different categories. Using the eye tracking recordings we could see where the users looked at, during the website interaction. In case our eye tracking had possibly missed any interaction we also tried to make use of possible memorization of security facts through the participants. For these we stored screenshots of the content of the websites the user had visited throughout the study and showed them to participants again after the first round of the study asking them several questions about the indicators. As a last resort we also did some logging of the interaction the user carried out whilst going through the tasks.

- **Eye Contact With the URL:** We used the eye tracking data to see whether or not a user looked at the URL during the website visit.
- **Eye Contact With the Security Indicators:** Again using the eye tracking data we looked at whether the participants eyes focused towards any security indicator within the browser.
- **Memory of URL:** When reviewing all websites we asked the participants whether they remembered the URL of the website they had been on, offering them four different choices: the original URL, a phishing URL and finally the possibility to state that they had no idea or to prose another domain name.
- **Memory of HTTPS:** This was nearly the same as the URL memorization questions instead that we asked for the encryption state of the website.
- **Time of Interaction:** Throughout the user study we logged the interaction time with each website to see whether that would change for our modified websites. We measured the time from the moment a website was loaded until it was dismissed again going back to the user study's overview screen.

As it was important to not prime our participants for security we hid the original purpose of the study until a debriefing at the end of it. Instead we announced the user study as being about the "visualization of websites". To keep the users distracted whilst viewing the eight different websites we also asked a couple of design questions that had to be answered

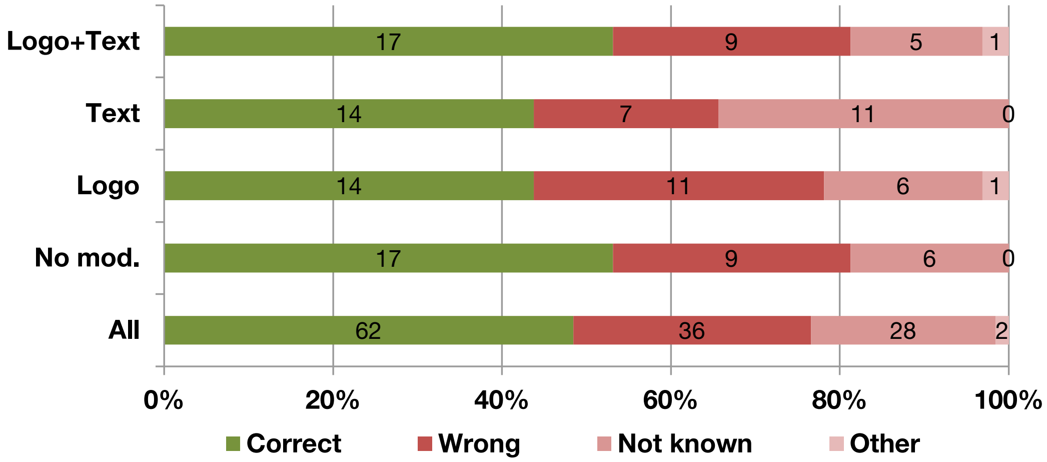


Figure 5.33: Correctness of domain guesses by the participants in total and for the different modification conditions. Participants were also allowed to state that they had no idea or could propose a different domain instead of the given two possiblities.

after viewing each website. We also instructed people to think aloud about their website interaction to see whether they would talk about the modifications towards the websites.

At the end of our study we debriefed the participants and asked them some questions about the overall concept using a final survey.

Results

We had 16 participants attending our user study – average age 23, mostly students – and hence repeated our Latin square balancing for the second half of the participants. 10 participants were male and none of the participants had taken part in a study about phishing before – we asked for that after debriefing the participants. Nearly all of them were students of IT programs. They stated to have a good average Internet knowledge: 4.4 (SD 0.7) on a 5-point Likert scale. They used the Internet for 5.3 (SD: 1.9) hours per day on average.

The first important aspect to look at, is whether our participants had noticed the different changes that we had introduced throughout our study. In general this worked out well. Having our think-aloud protocols, the users' questionnaire data and the eye tracking results we combined those different sources and found out that about 81% of the participants had noticed the logo changes and about 80% of the participants had noticed the text changes. However, this means that our changes still were overlooked by about 20% of our participants. We also had cases were the influence of the changed logo was so strong that the participant believed to be on the website belonging to the brand of the replaced logo.

Looking at how well the participants remembered the URLs or whether they remembered having seen a phishing website, the results suggest that our plugin did not help that people had a better memorization of phishing attacks or the URLs. On average the participants

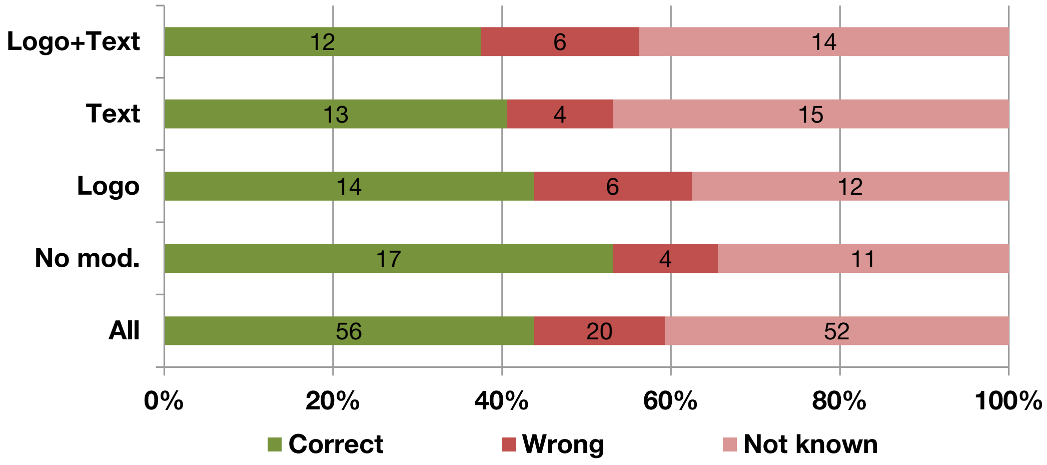


Figure 5.34: Correctness of encryption guesses by the participants in total and for the different modification conditions. Participants were also allowed to state that they had no idea about the encryption state.

chose in 48% of the cases the correct domain name which is close to guessing. We offered them the original and the phishing domain name to choose from but they were also allowed to state that they had no idea or they could propose a different domain name. In the Logo+Text but also in the condition without any modification 17 correct of 32 possible answers were given (53%). This shows that none of our modifications enhanced the user's URL retention compared to the baseline (see figure 5.33 for details). When asking the participants at the end of the study how much attention they had paid to the URLs they answered with 1.9 (SD 1.0) in average on a 5-point Likert scale (1-'not at all' to 5-'very close') confirming the retention results.

The results regarding the retention of the encryption state are even worse (see figure 5.34) and when being asked in the end how much attention the users had paid to security signs they answered with an average of 1.5 (SD 0.5).

This could be also confirmed looking at the data we recorded using the eye-tracker. Figure 5.35 shows the number of websites for which a participant had look at least briefly towards a given security indicator. We looked at different browser areas: the URL, the site identity indicator combined with the https-scheme indicator, the browser tab and the history return button. The fact that the participants' eyes came across a given indicator does not necessarily mean that they perceived and understood the indicator.

When finally asking our participants about the level of disturbance that was created through the different modifications, the textual modifications were rated to be more disturbing than the logo modifications (see figure 5.36). We also asked our participants about some of their feelings they had when using the plugin. From figure 5.37 one can easily see that the plugin confused and disturbed many of our participants while it made a lot less people think about safety.

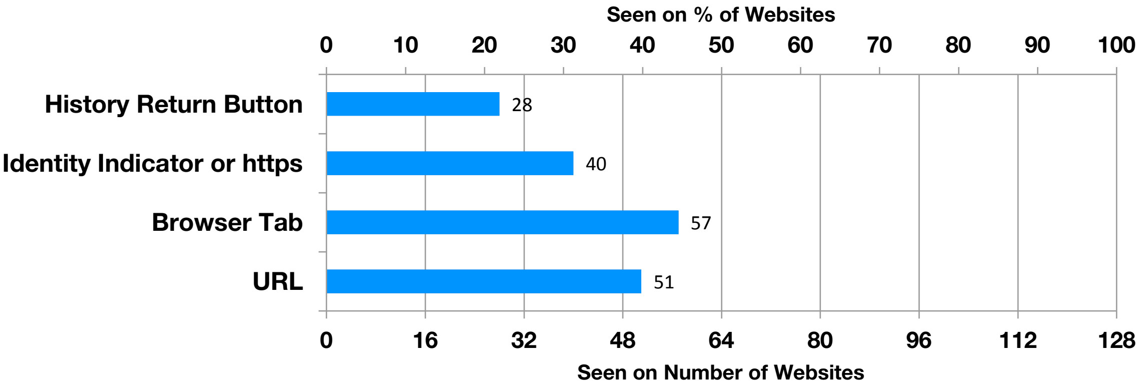


Figure 5.35: Number of security elements that have been looked at at least briefly by the participants on the 128 different websites shown according to the eye tracking data.

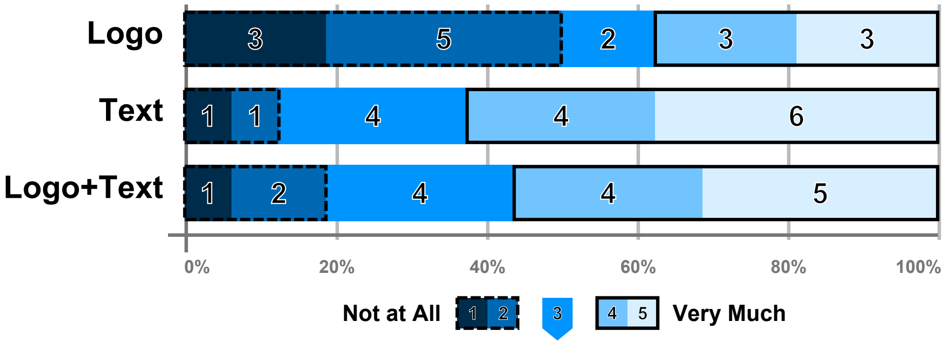


Figure 5.36: Level of disturbance experienced by the participants for the different types of changes.

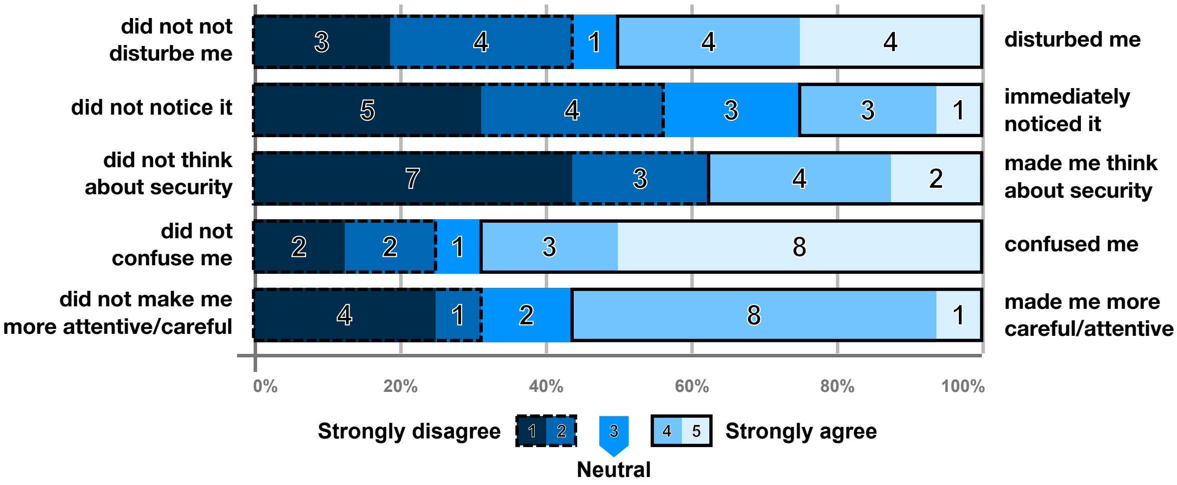


Figure 5.37: Different feelings of the participants towards the BrandTrustMinimizer concept.

5.7.5 Discussion and Limitations

Looking at our overall results our concept of getting the users focus away from the content area did not work in any way. No matter which kind of measurement was taken into account the users did not look more at other indicators. However, our modifications were noticed by most of the users and understandably they did not like them very much.

When developing the concept we originally had thought of a button within the browser chrome to acknowledge the changes of a website and revert them. Perhaps placing such an element next to the security elements would help a little, as users that are disturbed by the content changes could revert them when shifting their focus towards the security indicators.

Nevertheless we think that the results of this study show that lowering the trust the user set into the content area is not really possible. The only effect that seems to be achieved is that participants get confused and in some cases even thought they were at the website of the replacement logo instead of the original website. These effects seem to be much stronger than the shift of the users' focus towards the security indicators. We hence conclude that non-blocking security indicators need to attract attention by themselves and cannot rely that the users would in any case give up their content-focused behavior.

5.7.6 Research Results

Within this project the main goal was to find out if the user intervention methods and hence security indicators are taken into account by users more often if we try to reduce their trust into the content area of the browser. Although we were not able to successfully prove this, the project generated a lot of other interesting findings.

ID *What is User Intervention?*

User Intervention as the ability to protect people from dangerous security events – phishing in the case of this thesis – was also to some extent the definition that we used as a basis for this project. However, we did not use this as our main measurement metric. We did ask people whether they remembered the URLs they had been on, which would have been a necessary step in detecting the phishing attacks, but mainly looked out for the different security indicators that users looked at. This is only a necessary but not sufficient criterion for the user intervention to work.

IH *How can HCI be Used to Enhance User Intervention Mechanisms?*

Using HCI research as a background we wanted to approach the problem of security not being the primary course of action from a different perspective. As shown in subchapter 5.6 it is possible to achieve effects of non-blocking user intervention mechanisms that have been designed making use of HCI research and principles. In case of this project we wanted to see whether we could use the findings from HCI research to raise the general security awareness in favor of non optimized user intervention mechanisms by lowering the users' trust towards the content area of the browser. It seems as if this does not work out.

IM *How can User Intervention be Measured?*

In case of this study we used many different evaluation techniques to find out whether the security indicators received more attention than without our website changes. Using an eye-tracker we looked at the real eye movements of the users and were able to see pretty accurately whether the security indicators were focused for a longer time or not. Self-reported data of participants is usually off to a certain extent and if the participants are asked whether they looked at the security indicators there is always the possibility of a slight bias that users report to have looked at the indicators because they think that this is desired behavior. Instead of asking for this directly we tried to challenge the users' memory. In case they would have had noticed the security indicators and discovered a possible attack they would hopefully had remembered this. With our eight websites being within the standard memorability range of 7 ± 2 [197] this should have worked out. Despite that newer research suggests a memorability of only four chunks of information [58], this would have still been enough to remember our four phishing attacks.

In case an eye tracker is not available other computer metrics can be taken into account. Mouse cursor movements for example might be used to some extent as a replacement [46].

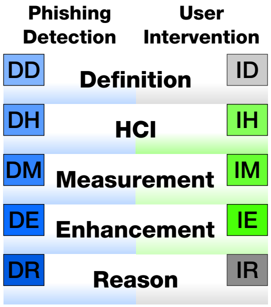
Using a within-subjects study for all different cases we only had two websites that used no plugin modifications and if our plugin had caused a general change in mind of our participants this would possibly also have effected the conditions that did not have any modifications. To make sure that such side effects cannot happen a between-subjects study should perhaps be preferred for future similar studies.

IE *How to Enhance User Intervention Quality?*

Throughout this thesis we used blocking, non-blocking and semi-blocking approaches for user intervention. As non-blocking security indicators are easily overlooked by the users, because security is never their primary goal, we tried to look at if this primary focus can be loosened in favor of security. In fact the website content seems to be so dominant towards the user that although we were able to confuse the users about the content to some extent these kinds of modifications to not create a focus shift towards security indicators. We conclude that security indicators have to attract the user's focus by themselves by delivering a high quality user intervention. Over time only the acknowledged advantages of such indicators can then lead to a more frequent access of users towards security interventions.

5.8 Visual Image Comparison For Phishing Detection and Reporting

This chapter is based on the work that was part of the bachelor thesis "Using Visual Image Comparison to Detect Fraudulent Websites" by the student Dennis Herzner [128] and the master thesis "User Interfaces for Indication of Visual Website Similarity for Fraudulent Websites" by Marc Mühlbauer [192]. Some parts of the project also led to a publication at the 30th SIGCHI Conference on Human Factors in Computing Systems (CHI2012) by Maurer and Herzner titled "Using Visual Website Similarity for Phishing Detection and Reporting" [185].



In the last subchapter we already saw the major importance of the content area of a website being more important to the users than the security indicators and tried to enhance this problem from a user intervention perspective. Within this chapter we want to turn the tables on this issue and want to use this fact as an advantage as well as an HCI parameter to develop a detector that is based on the visual look of the content area.

Most phishers try to make their phishing websites look the same as the original website they are impersonating which is in many cases done by simply copying large amount of the HTML content. If a website looks the same, a user will more easily fall for it. On the reverse, this also means that if a website looks the same it can be easily compared visually to other content images of websites. This is exactly the approach we want to take within this chapter.

Within the first part of the chapter we describe our development of a detector mechanism that uses visual comparison between screenshots of websites to find similar looking websites that are potentially impersonated. We compared many different image comparison methods within a huge detector test carried out using the test set presented in subchapter 5.1.

To make this HCI-based approach complete, we also co-develop a user interface concept for this approach that makes use of the visual comparison properties and other findings that were generated throughout this thesis. Finally this user interface has been evaluated in an interactive online study with participants from around the world.

5.8.1 Concept: Detecting Phishing Through Visual Similarity

Our general concept is based on the fact that due to branding and design not a single company website – or rather its exact graphics representation – exists twice. In case we would be able to find a website that does look exactly like an original website but is not affiliated with that website in any way, the other website will most certainly denote an impersonation attack. This project makes use of this effect in a twofold way: firstly, we can use it to design and evaluate a detection methodology that will yield similarity scores for unknown websites a user visits by comparing them to already known websites. Secondly, in case the

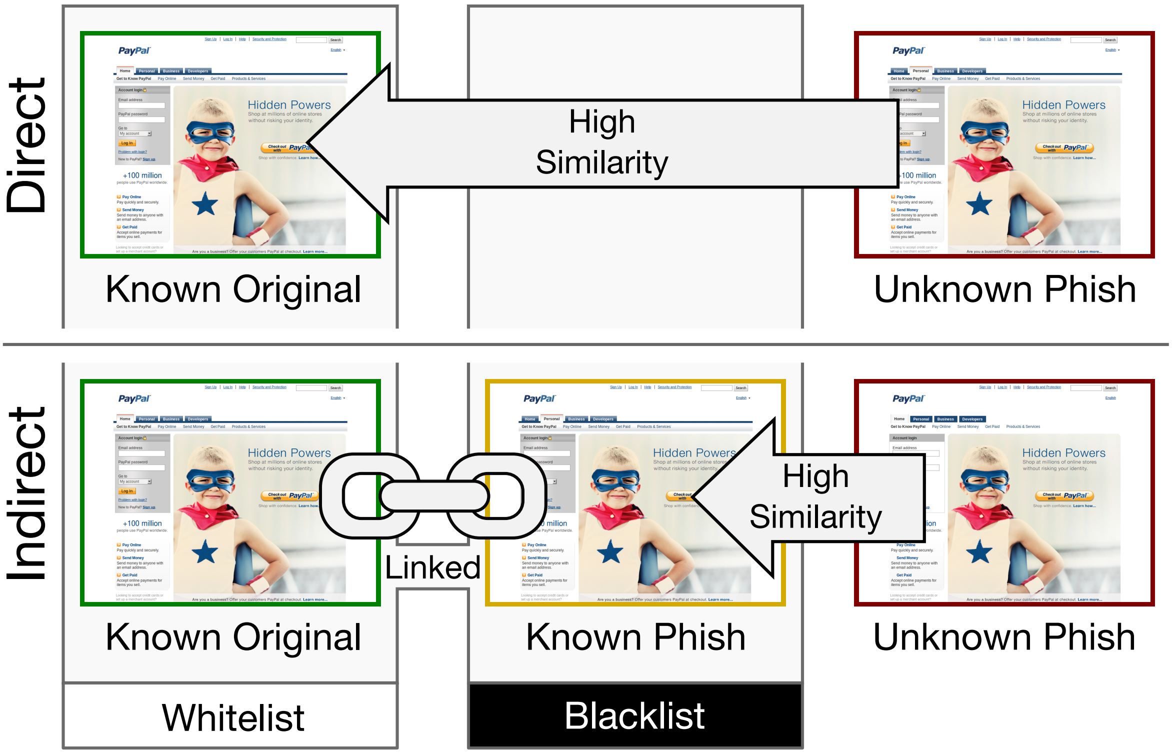


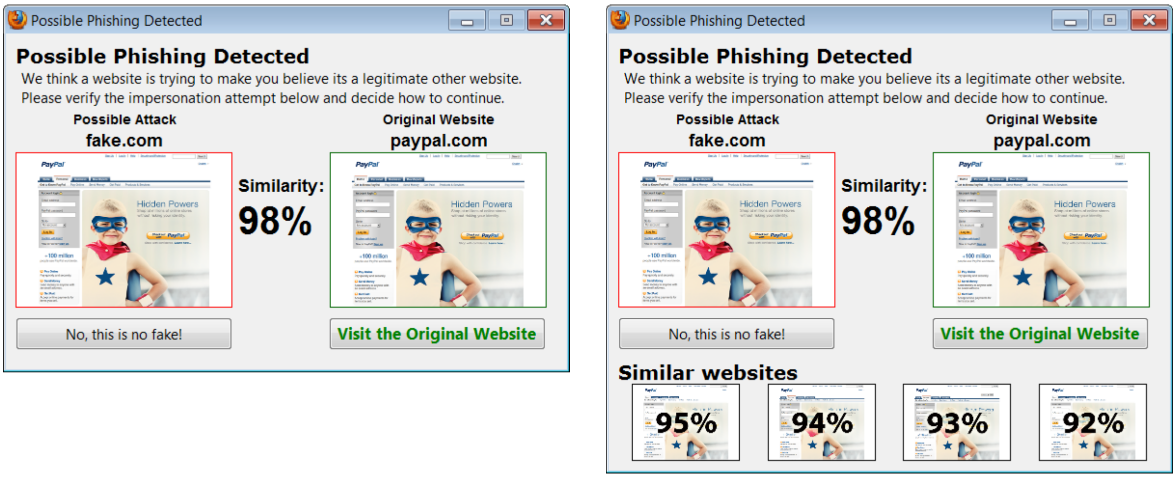
Figure 5.38: Examples for direct and indirect visual detection methods to find an attacking website either by looking at a whitelist of original websites or at a blacklist of known phishing websites.

impersonation is most likely, the visual similarity can be shown to the user and hence be used as means to create an understandable user intervention mechanism.

The Detection Process

The actual detection mechanism we imagine would work as follows: Whenever a user visits a website, a screenshot of the content area is generated and then compared against a huge list of known websites. This can be used for two different kinds of detection (see figure 5.38). The first way of finding phishing attacks is to "directly" compare an unknown website against a list of known original websites (whitelist). In case a high similarity is found, between two different URLs, the detector knows which website has been potentially attacked. The "indirect" detection method uses a blacklist of known phishing websites and searches for matches on this list. In case an unknown website is very similar to a known phish it will most likely be a copy of this attack. Linking the known phishing websites with their respective original website makes it again possible to know which original website has been attacked.

Comparing the visual similarity of websites might at first sound complicated and one might argue that HTML content based approaches would be much easier but the HTML content is only indirectly associated with the graphically rendered image. The problem that multiple



Possible Phishing Detected

Possible Phishing Detected

We think a website is trying to make you believe its a legitimate other website. Please verify the impersonation attempt below and decide how to continue.

Possible Attack fake.com

Original Website paypal.com

Similarity: 98%

Possible Phishing Detected

Possible Phishing Detected

We think a website is trying to make you believe its a legitimate other website. Please verify the impersonation attempt below and decide how to continue.

Possible Attack fake.com

Original Website paypal.com

Similarity: 98%

Figure 5.39: First draft of how the warning dialog of the user intervention method for visual similarity detection could look like [185].

versions of HTML code may lead to the same visual representation might be exploited by phishers to avoid detection (cf. phishing page polymorphism [158]).

Whatever kind of visual comparison is used, the detected elements are always returned with a similarity score. A user intervention mechanism has then to decide up to which threshold of the similarity score a similarity is non-critical. Similarities above this threshold trigger the user intervention warning.

Visual Similarity-Based User Intervention

Having the visually similar screenshots of websites, additional information about those websites and a visual similarity score, a lot of data exists that can be used to create a usable user intervention method. When thinking initially of this concept we already created conceptual warning designs that can be seen in figure 5.39. A general concept of the dialog design is to make use of the screenshots to express the general observation of two different websites being very similar in the warning dialog. The user can compare the screenshots and then verify which website she really wants to visit. As the screenshots change with every warning the dialog is also less prone to habituation.

5.8.2 Detector Architecture

Our detection process is carried out as a client-server-architecture. This is necessary as the data that needs to be checked is generated on the user side whilst the black- and whitelists of other websites take up large amounts of data and can hence not be stored on the client computer. We start by describing two different possible server architectures. Depending on the architecture the concept will by more or less privacy invasive at the cost of white-

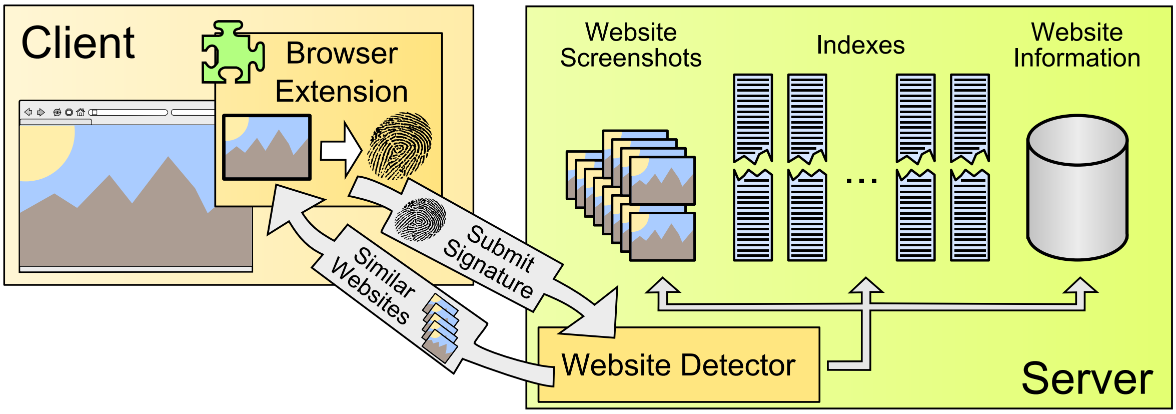


Figure 5.40: The client-server architecture of the more privacy-relevant version (based on [185]).

and blacklist quality. We will first describe the more privacy-relevant concept of the server before explaining the client architecture and talk about the privacy tradeoffs afterwards.

On the server side three different main components have to be available that are used by the detector component (see figure 5.40). The most important component is a number of different visual similarity indexes that contain image information depending on the visual similarity detector. These indexes are then used for quick lookups on image similarity. Although our concept would in general work with one image detector the server should be able to host multiple detector indexes. In case of our project we used these multiple detection indexes to compare the detector performance. In practical use, it would be possible to combine the results of multiple detectors to enhance the detector performance. Besides the image fingerprint data itself the indexes also store an ID for each website that can be used to look up additional website information that is stored in an extra database (second main component). This reduces the size of the indexes and ensures that additional website data is not stored multiple times for each image comparison index. A third component on the server that is actually not necessary for the pure detection process are the original screenshots that have been used to generate the comparison indexes. The website detection process would also work without those but the server would then be unable to transfer them to the client to be displayed in the user intervention dialog.

On the client side, a browser extension is used to carry out the detection process. This extension takes a screenshot of the website that a user is currently visiting and computes a fingerprint value out of the screenshot that can later be handled by the respective detector on the server side. It would also be possible to transfer the screenshot to the server and compute the necessary fingerprints on the server-side. Such an approach would protect secret fingerprint computation code but on the contrary this would take up much more bandwidth and would denote an extensive privacy threat as the screenshots yield much information about the user's current actions – as opposed to the hash-like fingerprints.

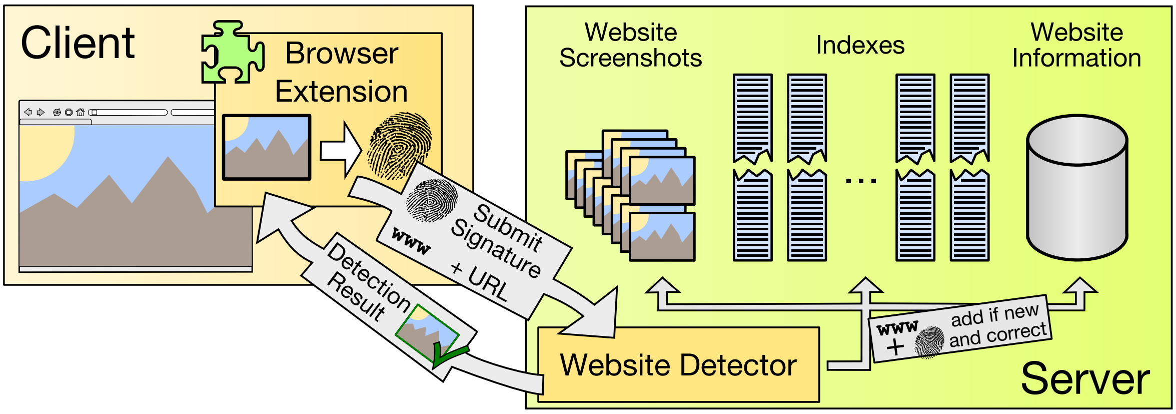


Figure 5.41: The client-server architecture of the more performance-optimized and less privacy-relevant version (based on [185]).

Once the fingerprint has been received by the server it can compute a list of visually similar websites and send those back to the client together with the respective screenshots, website data and similarity scores. On the client side the plugin can then compare whether the submitted fingerprint is nearly equal to one of the websites and in case a website matches but the URL is different from the original URL a warning can be displayed.

The big advantage of this architecture is that the user only has to submit the fingerprint of the website screenshot which makes it impossible for the server to track the user's website visits. A downside of this approach is that the check for an eventual matching phishing website has to be done on the client side and hence requires much more data to be transmitted back to the client.

In a less privacy-relevant version of our architecture (see figure 5.41) we would transfer the URL that the user is currently visiting together with the image fingerprint to the server. This has two major advantages. On the one hand the server can immediately reason about whether the visited URL might be phishing or not and include this result in its answer to the client. The second big advantage lies in the maintenance of images on the server side. In case the server receives a fingerprint for a website that is not yet known in the servers' database it can add the respective URL and fingerprint to its database for future searches. In case the URL is already known to the server and the fingerprint mismatches the one being stored on the server this could mean that the design of the stored website has changed and that it needs to be reindexed by the server. In both cases the server has to make sure that the submitted information by the clients can be trusted to not fall for attacks submitting faked fingerprints on purpose.

5.8.3 Evaluating the Detector

For the evaluation of our method we did not want to create our own image comparison algorithm as this is a huge research area in itself. Instead, we wanted to test the applicability of different image comparison methods towards our concept. Comparing website screenshots should in many dimensions be relatively easy compared to photo comparison in general as our screenshots are known to be mostly distortion free (e.g. not rotated, no clipped images, same camera angle).

Different Detectors Tested

Four our tests we used eleven different image comparison features that could be easily tested as we used a software framework for image comparison called LIRe [171] that has all eleven different features already on-board and can also be used to generate large image indexes for future image comparison that we needed on our server-side. The eleven features chosen for our evaluation are:

A first set of of three features have their foundations in the MPEG-7 standard for media description [50, 180]:

1. **Scalable Color (SCD)** is a color histogram using the HSV color space that characterizes the global color distribution of an image.
2. **Color Layout (CLD)** describes a spatial distribution of the YCbCr color values of an image by dividing it into 64 single regions.
3. **Edge Histogram (EH)** computes the spatial distribution of five different types of edges throughout the image.

Three other detectors use combined approaches and are hence called compact composite descriptors [43, 44]:

4. **Color and Edge Directivity Descriptor (CEDD)** combines a 24-bin color histogram together with texture-information.
5. **Fuzzy color and texture histogram (FCTH)** works similarly as the CEDD descriptor but uses a different type of color information gathering.
6. **Joint Composite Descriptor (JCD)** combines the information from CEDD and FCTH into one new. feature with even more texture information

Besides these descriptors we also use an **HSV and RGB**-based (7) histogram descriptor; a descriptor based on the **JPEG coefficients** (8) and their histogram and the **Auto Color Correlogram (ACC)** [131] (9) descriptor that makes use of spatial correlation of colors. The last two descriptors consist of one descriptor that is based on the **Gabor** [175] (10) filter and a descriptor that makes use of the textual features corresponding to human visual perception found by **Tamura** et al. [281] (11).

Test Set Recapture

As an input to our detectors we made use of the test set described in subchapter 5.1. It consists of 10,030 phishing URLs that have all been reviewed and assigned to one of 1,152 original websites. Removing phishing websites that could not be used for different reasons 3,603 verified phishing websites remained. 348 additional phishing URLs had to be neglected for this evaluation as they did not attack a specific brand. In such cases the phishing websites had no similarity towards a specific website and could only be detected by the "indirect" method.

Testing Methodology

When testing the performance of our detector we were mainly interested in the detection rate depending on the image comparison method used. In general this means the true positives (phishing websites detected) and the false positives (original websites accidentally detected as a phish). As an additional metric within our measurements we were not only able to determine whether a detector was able to correctly detect a phishing website but we could also check whether the website was matched to the correct parent website or brand – please see section 5.1.3 for an explanation of the differences between parent websites and brands. Using this brand matching as a metric we did several different measurements to find out about the detector performance: For the direct and indirect conditions we checked for each phishing website whether a lookup against the **list of phishing websites (indirect)** or **original websites (direct)** would return the most similar result from the same brand as the website tested. We also did a third test where we tested **half of the elements of the phishing list** (randomly selected) as a subset against an index consisting only of the other half of the phishing websites to see how the index size would affect the result. All those tests only gave us a number about the true positives that are achieved with each given comparison method. To get to know something about the false positives too, we had to use a **combined index of phishing and original websites**. In case of such a test the similarity threshold value is of great importance as depending on the threshold one could always achieve that every phishing website would be detected at the cost of having a lot of false positives. A good metric here is to find the equilibrium threshold at which the number of false positives and false negatives is equally high as this point can be taken as a standardized value to compare detector quality. It has to be noted that in some cases the query image has also been part of the test set when querying some indexes. In such cases we ignored this first result – as it was always a perfect match – and used the second search result for our measurements.

Besides all those detection performance measures we also tested other parameters that play an important role when building such a system. These were: the **average time needed to create a search index entry** from a given screenshot; the **index size** an added entry had on average and finally the **time a similarity query** towards the index took. When adding images to an index we always provided a full resolution screenshot to the indexing function of LIRe. If the image descriptor needed a smaller resolution of the images the computation of this image was part of the insertion time measured. All our computations were done on

Windows 7 machines having an Intel i7 860 CPU and 8 GB of RAM with no other major tasks running in the background.

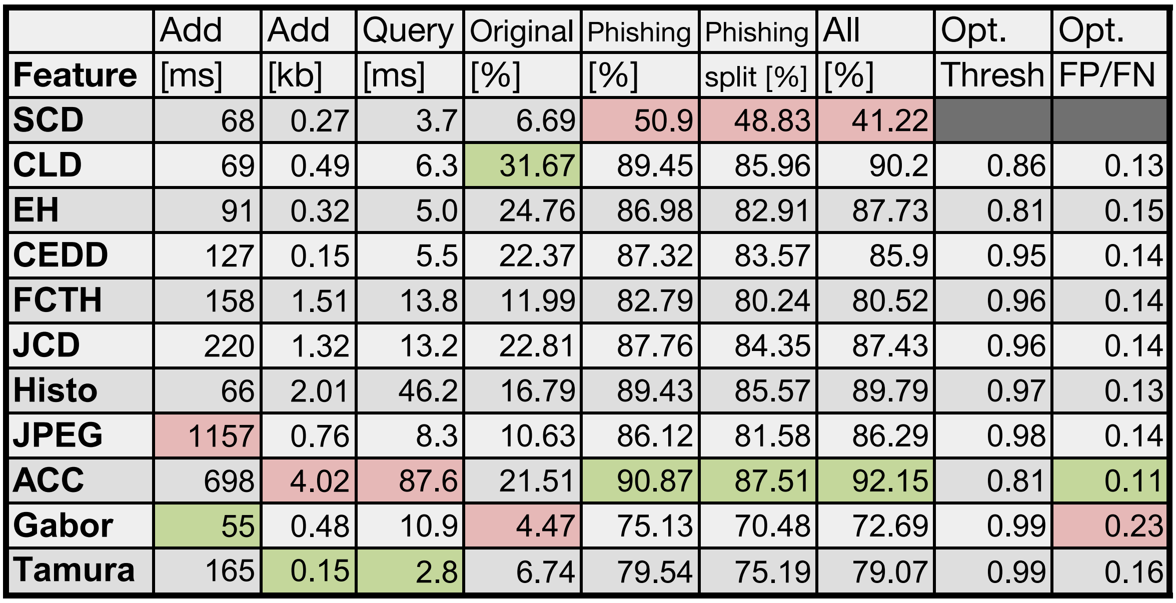
Detection Results

A huge problem that we faced when doing our detection was that LIRe does not return a similarity score within a predefined range (e.g. from 0 to 1) but instead returns a distance value of the compared images that is dependent of several parameters of the image comparison method. A distance of 0 denotes two equal images in the reference system of this image comparison method. A maximum distance value does not necessarily exist as this can be dependent of many factors. To still be able to compare the results of the different detectors against each other we chose to normalize our detection results. To do so, we calculated all distance values for our different indexes and measurements. Afterwards we identified the maximum distance value for each feature and used it to assign a final similarity score by dividing each distance by the maximum of all distances and subtracting the result from 1: $\text{similarity}_{feature}(i) = 1 - \frac{\text{distance}_{feature}(i)}{\text{max\_distance}_{feature}}$. After doing this we had a similarity score ranging from 0 (not at all similar) to 1 (equal).

Looking first at the results besides detection performance, the different image descriptors performed very differently. Whilst an average addition of an image to the Gabor index took only 55 ms on average addition to the JPEG coefficient index took more than a second. When adding thousands or millions of entries to an index this may play an important role. Looking at the size of the inserted entries Tamura and CEDD outperformed all other detectors having in average only 0.15 KB of data per image in the index while the Auto Color Correlogram needed more than 4 KB per indexed item. Querying the indexes seems to be very related to the size of the index as Tamura was again fastest here (2.8 ms/query) and ACC again slowest (87.8 ms). The results for all detectors can be found in the first three columns of table 5.15.

The rest of table 5.15 shows the results of our different performance tests using different features. Testing all phishing websites against an index containing the original websites the color layout descriptor performed best, finding the correct parent website for 31.7% of all phishing websites (Gabor was worst). Comparing the phishing websites against the phishing/blacklist index using the indirect approach, the ACC feature returned a phishing website of the same brand in 90.9% of all cases. Using half of the phishing websites as a potential blacklist it still achieved a performance of 87.5%. When using an index containing both types of websites (phishing and original) the ACC feature returned a website of the same brand in even 92.2% of all cases.

All these values looked at the most similar of all websites that were returned without taking the exact similarity scores into account. We hence tested only phishing websites against those indexes. To get a real comparison about how the false positives and false negatives would look like, one has to look at the detection rates dependent of the similarity scores. Using a threshold of 0.81 using the ACC feature one could achieve an equilibrium where 11% of all phishing websites would stay undetected and 11% of all original websites would cause a wrong phishing alert. The false positive and false negatives values according to the



	Add	Add	Query	Original	Phishing	Phishing	All	Opt.	Opt.
Feature	[ms]	[kb]	[ms]	[%]	[%]	split [%]	[%]	Thresh	FP/FN
SCD	68	0.27	3.7	6.69	50.9	48.83	41.22		
CLD	69	0.49	6.3	31.67	89.45	85.96	90.2	0.86	0.13
EH	91	0.32	5.0	24.76	86.98	82.91	87.73	0.81	0.15
CEDD	127	0.15	5.5	22.37	87.32	83.57	85.9	0.95	0.14
FCTH	158	1.51	13.8	11.99	82.79	80.24	80.52	0.96	0.14
JCD	220	1.32	13.2	22.81	87.76	84.35	87.43	0.96	0.14
Histo	66	2.01	46.2	16.79	89.43	85.57	89.79	0.97	0.13
JPEG	1157	0.76	8.3	10.63	86.12	81.58	86.29	0.98	0.14
ACC	698	4.02	87.6	21.51	90.87	87.51	92.15	0.81	0.11
Gabor	55	0.48	10.9	4.47	75.13	70.48	72.69	0.99	0.23
Tamura	165	0.15	2.8	6.74	79.54	75.19	79.07	0.99	0.16

Table 5.15: Performance measurements and detection rates for the eleven different image features. From left to right: average time of adding an image to an index; average space needed by an image in the index; average time taken for performing a similarity query; percent of phishing websites correctly assigned when testing against the index of original websites (direct); percentage when testing against the phishing index (indirect); percentage when splitting the phishing test set into halves; percentage when testing against phishing and non-phishing; equilibrium threshold and resulting false positive/negative value at this threshold. The best value of a given test is colored green the worst red.

moving threshold can be found in figure 5.42. The same results plotted as a ROC curve can be found in figure 5.43. Please refer to section 1.4 for an explanation of the diagram types.

Detector Discussion

Looking at the performance values only, Auto Color Correlogram (ACC) outperformed the other image comparison features in most cases. Only in case of the original websites it ranked fifth. Taking the non-detection parameters into account ACC was by far the slowest detector and took double the time of second slowest detector (Histogram). Combining those findings with the findings of the Color Layout Descriptor one can see that this one performed quite well in terms of speed and was even ranked best when checking against the original websites. In all other detection domains CLD was only a little worse than the ACC image comparison. We would hence recommend to use the color layout descriptor (CLD) for practical use.

Looking at the false positive and false negative thresholds an equilibrium rate of 11% is not acceptable for a practical field deployment. Missing one out of 10 phishing websites would be critical but still somehow okay, but seeing an erroneous warning at every 10th website

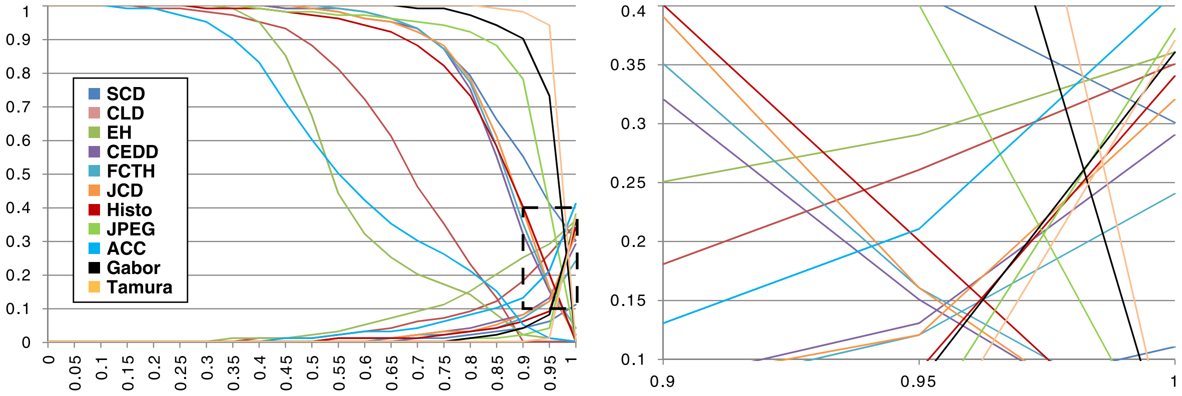


Figure 5.42: False positive and false negative percentages (y-axis) according to the the varying detection threshold (x-axis) for each of the different image comparison features.

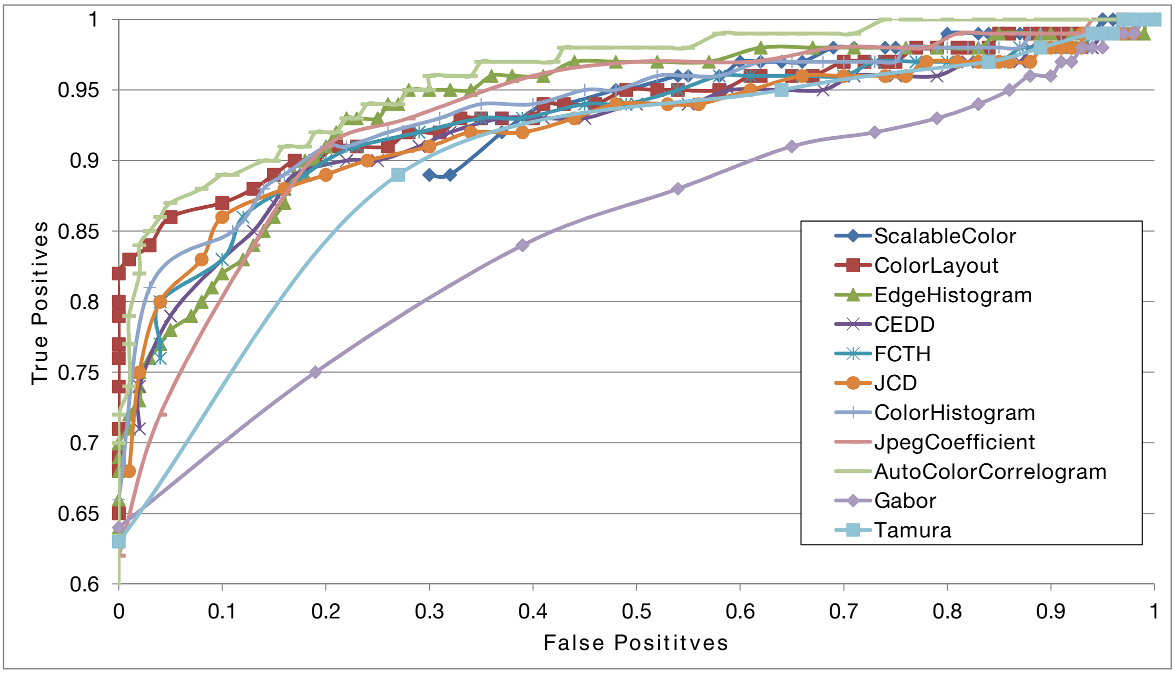


Figure 5.43: True and false positive ratios of the different detectors depending on different thresholds plotted as a ROC curve.

would be too much. By shifting the thresholds it would get possible to reduce the number of false positives – original websites denoted to be phishing – close to zero when using a threshold of approximately 0.5 (see figure 5.42). In return this would mean for most detectors that they also would miss nearly all of the phishing websites. Only the JPEG and ACC based detectors would still work to some extent. We believe that the problem here arises due to our test set missing important images. Although we already tried to test everything with a large test set, it is simply not enough to have only one screenshot of each original website in the database. Login pages, start or landing-pages of websites look often a lot different from the other pages of a website. We argue that in case one would increase the size of this testing environment to a real world sample the approach would lead to remarkably better results. When we looked at undetected phishing images in detail we saw that for a lot of attacks only a specific screenshot was missing on either the black- or whitelist. Since the results from our queries to the indirect indexes showed a high detection potential it would make sense to use only this kind of detection in case no proper set of original websites is available as a whitelist.

Another severe problem to this detection methods are websites that have elements that change often. The Microsoft Bing[19] search engine is one such example. It uses a full screen background image that is changed every day. In such a case an image-based detector would not work. Websites containing videos or other time-dependent media would also be a problem as the time the screeenshot is taken would influence the detection result. For such cases one could use masking to filter out certain image regions before adding an image to the database and reusing the same mask whenever comparing those images.

Another option to raise the detection results could perhaps arise from the combination of different features or by the use of a feature specially crafted for website comparison.

Using the LIRe framework for the detection process the image comparison is still done against all fingerprints of all images in the index. Even for our test set with over 5,000 images the results were still available within milliseconds but the query time would rise linearly ($O(n)$) as the number of stored images rises. Other practical projects like the Google image-based[20] search show that this can be done fast even with millions of images in a database. If the structure of the image feature itself is known, it can be possible to arrange the image fingerprints in a tree-like fashion and bring down the search time for a single image or multiple similar images down to a tree lookup time that is usually $O(log(n))$.

The architecture that has been presented within this project would easily support all those changes as the detection component can easily use other features or even be replaced by another query method.

[19] `http://www.bing.com`

[20] `http://google.com`

5.8.4 User Intervention Design

All prior efforts made to detect possible similar websites are useless if the results cannot be properly reported towards the user, using a user intervention method. Given the visited website, a similar website and its similarity score a lot of information exists that can possibly help the user to take a final decision of whether or not she was tricked into visiting an impersonating website. A first interface draft for our concept was already developed at the beginning of the project and resorted to suggestions about designing usable security interfaces (see figure 5.39). To develop a final user intervention methodology we also conducted a focus group to refine the design ideas before creating two final warning designs that were evaluated afterwards.

Focus Group

In a focus group consisting of six participants (two female, all HCI students) we wanted to find out which expectations these users would have towards such a kind of user intervention, based on visual similarity of websites.

We started to discuss their general feelings about browser warnings with them before asking them how they judge websites and which metrics they use when looking at similar pictures. In case of websites professionalism, the layout, colors and fonts played an important role for our participants.

After presenting the concept of our idea, the participants were asked to create their own warning draft in the second phase of the focus group. We had not shown them any of our previous drafts up to that point and only provided them with diverse kinds of pen and paper material to craft their designs. After finishing the design phase we showed our first drafts and discussed pros and cons of all designs.

The popup style of our drafts was disliked by most of the participants as they felt annoyed by appearing popups no matter what kind of information was contained in such a window. They still wanted to see less text and were unsatisfied that the dialog's confirmation options were of equal importance. They wanted to have a dialog design that would make it harder to choose the unsafe option without annoying users too much. A few participants of the focus group found it confusing to have two screenshots visible within the warning window at once.

Final Designs

Taking all the findings from the focus group into account we finally created two different warning designs for our user intervention method. The main difference between both interfaces was that one showed the screenshot of the currently visited website at first sight while the second design hid that screenshot in a pop out section and users saw only the safe option in the beginning. Figure 5.44 shows these two different final designs. The final dialog that we created is not a popup dialog as in our first drafts but is instead loaded in the content area of the browser instead of the website that should actually be loaded. Using a large red backdrop the warning can immediately be recognized. In the first version of our dialog (A) we show the screenshots of the website the user has tried to reach and the website that is most similar to that one side by side. As a safe option the found original website is shown larger with a green bar and a checkmark attached to it. The website that is suspected to be phishing is shown smaller, semi-transparent and with an information icon to state that this is not the suggested option. To acknowledge the change of plans and switch to the safe website the user only needs to click once on the suggested thumbnail. If she wants to stay on the current website two clicks are needed. The first click on the semi-transparent option enlarges the screenshot and makes it 100% visible. With a second click this option can finally be chosen. In case of our second design (B) we only show the suggested original website at first glance. To ignore the warning the user has to open up a hidden part of the dialog – that contains additional information – where she can find the screenshot of the current website. In this case two clicks are again needed to override the warning when not choosing the safe option.

For both warnings we also needed to create a dialog depicting the "indirect" case in which the original has been found because the phishing website looks similar to another already known phishing website from a blacklist. In those cases we showed both phishing images next to each other. The similarity score is placed differently in such a case as the similarity is measured between the two phishing websites and not between the original and the phishing website.

5.8.5 User Intervention Evaluation

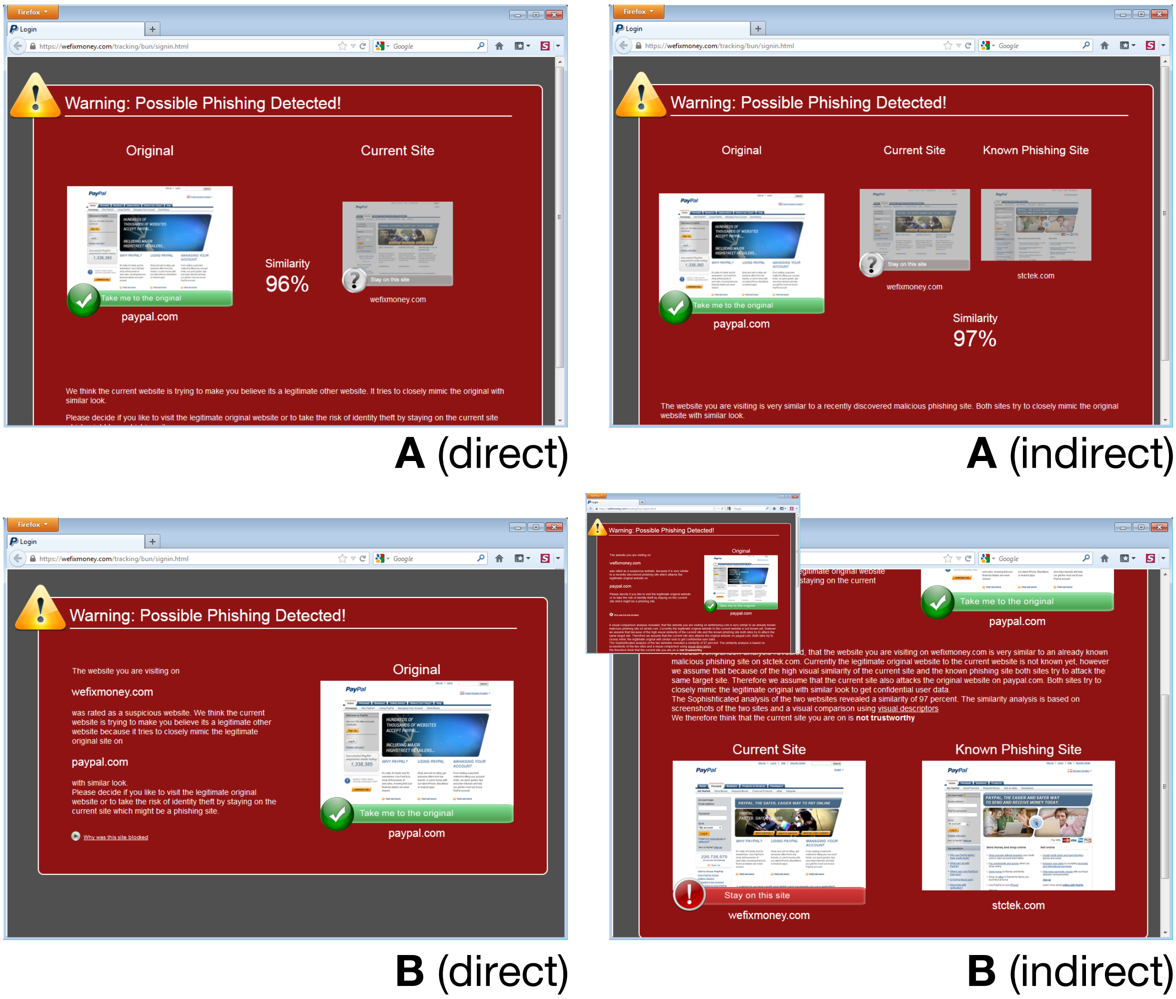
To evaluate this user intervention methodology we wanted to not only find out which version of our dialog would be better but also measure how well the concept of the visual similarity comparison would help people to decide correctly they should stay on a misclassified phishing website (false positive) or change towards the original website in case they landed on an impersonation attempt (true positive).

Evaluation Methodology

We did not use a lab or field study in the surroundings of the university this time but tried an interactive virtual online study using Amazon Mechanical Turk[21] instead. We prepared a within-subject experiment with three different independent variables. The dialog version (A or B) was the first independent variable. Besides this, we had direct or indirect phishing detection that was tested and finally a warning could be true positive or could have appeared in error (false positive). These three independent variables with two levels each, led to a total of eight different test cases.

As dependent variables we measured the final decision a user took when seeing our warnings and a lot of other parameters, like the clicks that were carried out, or the time frame

[21] http://www.mturk.com



We think the current website is trying to make you believe its a legitimate other website. It tries to closely mimic the original with similar look.

Please decide if you like to visit the legitimate original website or to take the risk of identity theft by staying on the current site

The website you are visiting is very similar to a recently discovered malicious phishing site. Both sites try to closely mimic the original website with similar look.

The website you are visiting on

wefixmoney.com

was rated as a suspicious website. We think the current website is trying to make you believe its a legitimate other website because it tries to closely mimic the legitimate original site on

paypal.com

with similar look.
Please decide if you like to visit the legitimate original website or to take the risk of identity theft by staying on the current site which might be a phishing site.

The website you are visiting on

wefixmoney.com

was rated as a suspicious website, because it is very similar to a recently discovered phishing site which attacks the legitimate original website on

paypal.com

Please decide if you like to visit the legitimate original website or to take the risk of identity theft by staying on the current site which might be a phishing site.

A visual comparison analysis revealed, that the website you are visiting on wefixmoney.com is very similar to an already known malicious phishing site on stctek.com. Currently the legitimate original website to the current website is not known yet, however we assume that because of the high visual similarity of the current site and the known phishing site both sites try to attack the same target site. Therefore we assume that the current site also attacks the original website on paypal.com. Both sites try to closely mimic the legitimate original with similar look to get confidential user data.
The Sophishticated analysis of the two websites revealed a similarity of 97 percent. The similarity analysis is based on screenshots of the two sites and a visual comparison using visual descriptors
We therefore think that the current site you are on is **not trustworthy**

Figure 5.44: The two final designs of our warnings derived from the findings of the focus group. Version B does not show the current website at first sight an instead hides it in an additional section of the warning. Both warnings had two different versions to illustrate the possibilities of direct or indirect phishing detection.

our warning was visible before it was dismissed. To show our warnings to users online, it was not simply possible to install the complete plugin within their browsers. Instead, we created a virtual browser window within the users browser that was then used to display the warning screens and the websites. The browser was not fully interactive. Instead, the browser chrome was created by using a screenshot of a browser looking just as if the user would be visiting a certain URL. All security indicators and the URL looked as if the user would be visiting a real website. For the content area of the virtual browser we also used a screenshot of the whole content of the respective website. With such a setup the participants could scroll through the content in case it was larger than the browser window. However, the virtual browser itself did not allow any interactivity (like clicking on links or buttons). It had a size of 1023x766 pixels and hence fitted completely in the browser window of users having resolutions above 1280 by 1024 and even on 1024 by 768 users were still able to experience the virtual browser window if they set their own browser to fullscreen. We monitored the participants' browser resolutions and did only display our virtual browser if their browser window was large enough to fully display our virtual browser. In other cases we displayed instructions about how to enlarge the browser window and what resolution would be necessary.

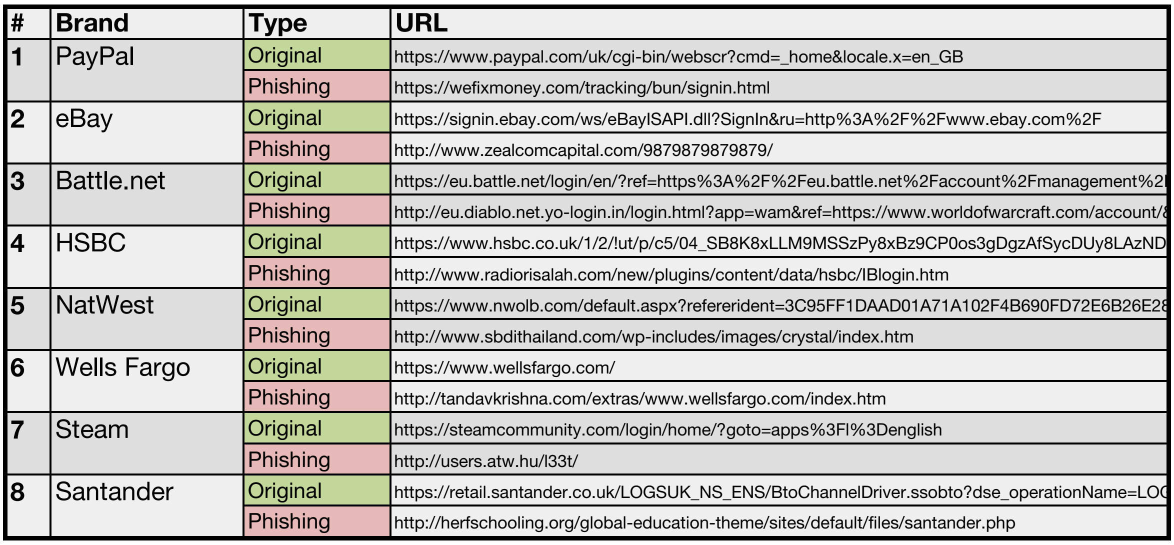
After a user had seen the virtual browser pointing to one website for a total of four seconds we opened our warning dialog. As we saw with our detector evaluation, the whole detection process using our client-server architecture could easily take a small while which was simulated by that. The warning message was fully interactive as it would be in a final browser implementation and stayed in front of the virtual browser until it was dismissed by either leaving to the recommended website or by choosing the icon of the current website to stay on that website. After having chosen one of the options the participants received a confirmation code they needed to go on with the study and had to answer two free text questions about what they experienced and why they took their decision.

We presented our eight different cases in random order using eight different websites for testing that are regularly suffering from phishing impersonation attacks (see table 5.16). Browser images for a phishing and an original state of all websites were used. In case the user was on a phishing website the only safe option to choose was the original website. In case of an original website our warning appeared in error and hence choosing to stay on the website would have been the more logical option. However, in any case choosing the safe option of changing to a trustworthy website would never denote a risk for the user. In such error cases we randomly picked another website that was proposed together with a low similarity score.

As always we hid the original purpose of the study and advertised it to be about "The Future Web Browser". We only asked our participants to use an interactive demo of a new webbrowser the same way they would, when browsing the Internet for themselves. Prior to the real tasks we always displayed a first test task showing google.com as a website with a standard popup that appeared that had to be dismissed. We did this so our participants could get used to the interactive browser setup.

A major difference to other studies was that we did not set a goal for the participants that they should achieve. In prior projects we used our "grandma is ill" scenario for example to justify why people should carry out some tasks online. Instructing people directly to carry out a task is often seen as forcing them to deliberately ignore security problems [271]. Within this study people did not know whether their goal was to actually stay on the current website, so they had to reason about it when seeing our dialog.

The whole study process was guided by a survey that started with some demographic information questions and finally debriefed the participants and asked questions about the concept and the different designs.



#	Brand	Type	URL
1	PayPal	Original	https://www.paypal.com/uk/cgi-bin/webscr?cmd=_home&locale.x=en_GB
		Phishing	https://wefixmoney.com/tracking/bun/signin.html
2	eBay	Original	https://signin.ebay.com/ws/eBayISAPI.dll?SignIn&ru=http%3A%2F%2Fwww.ebay.com%2F
		Phishing	http://www.zealcomcapital.com/9879879879879/
3	Battle.net	Original	https://eu.battle.net/login/en/?ref=https%3A%2F%2Feu.battle.net%2Faccount%2Fmanagement%2
		Phishing	http://eu.diablo.net.yo-login.in/login.html?app=wam&ref=https://www.worldofwarcraft.com/account/
4	HSBC	Original	https://www.hsbc.co.uk/1/2/!ut/p/c5/04_SB8K8xLLM9MSSzPy8xBz9CP0os3gDgzAfSycDUy8LAzND
		Phishing	http://www.radiorisalah.com/new/plugins/content/data/hsbc/IBlogin.htm
5	NatWest	Original	https://www.nwolb.com/default.aspx?refererident=3C95FF1DAAD01A71A102F4B690FD72E6B26E2
		Phishing	http://www.sbdithailand.com/wp-includes/images/crystal/index.htm
6	Wells Fargo	Original	https://www.wellsfargo.com/
		Phishing	http://tandavkrishna.com/extras/www.wellsfargo.com/index.htm
7	Steam	Original	https://steamcommunity.com/login/home/?goto=apps%3Fl%3Denglish
		Phishing	http://users.atw.hu/l33t/
8	Santander	Original	https://retail.santander.co.uk/LOGSUK_NS_ENS/BtoChannelDriver.ssobto?dse_operationName=LO
		Phishing	http://herfschooling.org/global-education-theme/sites/default/files/santander.php

Table 5.16: A list of the eight different brands and the respective URLs used within the user study to evaluate the user intervention concept for visual similarity. For each brand an original URL and a phishing URL was needed as the brands were randomly used as phishing or non-phishing between the participants.

Evaluation Results

We had 50 turkers (the Amazon Mechanical Turk users) that finally completed our study (in case someone dropped out mid-way through we recruited another person). 18 participants were female. Most participants came from the US and India with an average age of 31 years (20 to 64). In average they used the Internet for 7 hours per day and had diverse occupations. They stated to have good Internet knowledge (average 4.3 SD 0.8) on a Likert scale from 1-“strongly disagree” to 5-“strongly agree”.

As each participant had to take eight decisions of staying or leaving on the website we collected 400 decisions in total. 277 decisions could be classified as correct (meaning the participants chose the option that was best for the given case). This leaves us with 123 “incorrect” decisions that have to be split again as only cases where people deliberately wanted to stay on a phishing website were really problematic. We only had seven of those (3.5% of the incorrect decisions). In all other cases people chose to change the website to the one suggested although staying would have also been safe and would have made more sense.

In nearly all phishing cases (97%) the participants correctly chose the suggested option to change the website but even more interesting is that they still chose the discouraged option in 42% of the error cases of our warning. Despite we had not provided the participants with the goal to navigate to a certain web page they still were able to reason about the current situation of the warning correctly in many cases and did not blindly click on the preferred option.

For the phishing cases we did not find a difference between version A (97% correct) and version B (96% correct). In case of the error cases version A seemed to work better (49% correct) than version B (35% correct). Direct or indirect warnings did also not play a big role (correct decisions: 69.5% direct vs 69% indirect).

Looking at how long the dialogs stayed open the decisions using version A were usually done faster (19.9 seconds average vs 21.7 seconds for version B). Incorrect decisions were also done faster due to some participants that had seemed to have skipped through the questionnaire as fast as possible.

Compared to the data type based study in subchapter 5.5 a lot more people unfolded the hidden section in our dialogs this time (103 times in total; 68 times for version B; 35 times for version A). The higher number for version B results from the fact that in those cases this section was needed to skip the warning.

Asking people after the debriefing for their favorite dialog style 68% favored version A. Considering the very good measurements of version A we had actually expected an even higher number. One participant that chose A over B put it like this: "[Using A one can] see the problem and does not have to read about it." Version B was seen as being less complicated and drawing the users attention more towards the safer option.

Analyzing our results statistically using the Cochrans Q-Test for within-subject tests with binary outcomes, the detection measurements differed significantly ($Q = 150.7, p < 0.001$). With a McNemar posthoc test (Bonferroni corrected) we found that the dialog version ($\chi^2 = 5.30, p = .030$) and the provided correctness of our detector ($\chi^2 = 99.7, p < .001$) were significant whilst direct or indirect detection had no significant influence.

As a last evaluation step we looked into the reasons for the incorrect decisions of our participants by classifying the textual reasons they had given after each decision. 21% of the 123 incorrect answers had to do with participants being in a hurry. One comment said for example: "just clicked the green option to get the confirmation code". We did not remove those participants from the results as we thought that real users could also behave in such a hurrying way. Fear was the biggest reason for wrong decisions (26%). The participants sometimes were so convinced of our browser mockup that they "chose the safe option to not get any problems". In 11% of the wrong decisions of version B the participants did not find the option to stay on the website that they were actually looking for.

5.8.6 User Intervention Discussion

The use of visual similarity as an explanation for impersonation warnings seems to work very well. Although we did not provide a specific goal to our participants they were in many cases able to find the situations when it did not make sense to follow the prominent option.

Although our study used eight repeated exposures to relatively similar looking warnings the average time a warning was opened did not decrease throughout the study and was always approximately 20 seconds. This confirms that the warning generates low habituation.

Within most measurements the warning design A showing all screenshots next to each other outperformed the design B. In some cases the hidden option to stay on the current website was not found by our participants in version B. Although some participants mentioned that version A could confuse people more easily we did not find any hints for that when looking at the interaction times or the number of correct and incorrect decisions.

5.8.7 Research Results

Within this project we looked at an overall concept for phishing detection and user intervention using the visual content similarity as means for both. Especially the user intervention concept worked very well and the participants were even able to handle simulated error cases quite well. In case the detector accuracy could be increased more by using different detectors or a larger screenshot set, such a kind of phishing detection and user intervention will definitely be very understandable and robust.

DD *What is Phishing Detection?*

Within this project we used the very classical approach of measuring false positives and false negatives when detecting phishing websites. To help the users in understanding the attack more closely and guide them towards a safe exit an additional parameter can be used in the phishing detection, namely the brand of the impersonated party. If a detector is not only able to distinguish between malicious websites and original websites but can also find out what original website is associated with each attack it becomes possible to automatically redirect the user towards the intended goal.

DH *How can HCI be Used to Build Detectors?*

As seen many times in this thesis before, the visual channel and the content area of the browser is the most important means of legitimacy judgments for Internet users. This makes it not only a perfectly suited property for user intervention but it can also be used for the detection process. On the opposite having the same means for detection as for reporting makes it possible to easily align the user's conceptual model and the implementation model [54]. As security is always a construct that is hard to understand this kind of model alignment can be very helpful when developing security measures.

DM *How can Detectors Be Evaluated?*

Instead of evaluating let alone our detection concept, we tested a variety of different image comparison features for their suitability towards detecting phishing websites. In case of this project the false positives and false negatives were not the only sensible measurement that could be taken and even in case of these measurements the interchangeable threshold played an important role. Knowing the exact brands that were assigned to each of the test websites it became possible to measure how well the brand recognition did work using a whitelist for direct comparison and a blacklist for indirect comparison.

Besides these measurements about the detection quality other metrics can also play an important role for the evaluation of a detector. How long does it take to generate the image detection indexes? How much space do these need depending on the detector? How long does a query to such a detector take? Although these questions might be somehow secondary at first sight, phishing detection is a time critical process and could not be used in case a detector would take several seconds for the detection.

DE *What Kind of Detection Works Best?*

Within this project we were not able to prove that visual similarity based detection is evidently the best working detection method so far, but it has several properties that make it stand out of other detectors that have been developed. By using the most important criterion for a user to judge a website it gets possible to use detector information directly to inform the user about the result in a way that is easily understandable. Another advantage of such an approach is that there are no technical means an attacker could take to adapt to the detection process without also lowering the impersonation quality of his attack.

DR *What Detection Overhead and Thresholds are Reasonable?*

For each of our detectors we calculated an equilibrium threshold that could would lead to an equal number of false positives and false negatives. These values were very high for all detectors. A false positive that would appear on every 10th website would greatly disturb the user during his browsing routines. Even when changing the threshold towards less frequent appearing false positives the number of correctly detected phishing websites also drops to a large extent.

For detector comparability the equilibrium threshold is actually a good measurement but normally it should not be used as a threshold for the user intervention method. The user intervention method itself should always try to minimize the number of errors that appear on original sites to a minimum even if that means that a few more phishing websites stay unreported. If too much habituation towards a user intervention method occurs it can easily happen that none of the interventions will be regarded anymore by the users rendering it completely useless even in the correct cases.

We suppose that by using a more complete data set and a better detector component these values can be easily brought down and in combination with other means like URL whitelists the computing and error overhead could also be reduced easily.

IH *How can HCI be Used to Enhance User Intervention Mechanisms?*

The visual channel and the content representation are the important properties when a user judges the authenticity of websites. Using these parameters to explain possible security flaws can hence greatly enhance the understanding of warnings. Within this overall approach we used these parameters for both the detection of phishing websites and the user intervention method.

IM *How can User Intervention be Measured?*

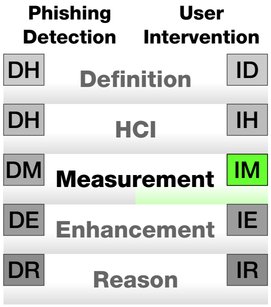
Within this project we did not conduct a local lab or field study but performed an online study using Amazon Mechanical Turk instead. With a virtual browser that was displayed within the participants' own browsers we tested how people would react to the displayed warning messages. To make such an online test feasible we had to use a more abstract browser representation than we would have been able to use in a real world study. However, the warning messages themselves were not affected by that in any way. Whenever possible the browser interactivity should not be reduced as this limits the users' access to browser functions they might have eventually used to resolve the security issue. As we were exclusively interested in the user behavior within our warning screens such a kind of study became possible and hence yielded a lot of other advantages. Using an online study a large base of test users can be quickly recruited. Being Internet users from around the world they denote a proper sample of the target audience for such a test.

IE *How to Enhance User Intervention Quality?*

For a user intervention method to be successful it is important that the users can understand the problem that has occurred to make them able to take a correct decision. Using a metric that is well understood by them – visual similarity in this case – is a great example for a well working user intervention method. Another important design decision for better quality is to offer the safe option of a dialog more prominently without making it too complex to ever reach it in case of errors that could appear. In our case a first click plus a short time for an animation to complete was needed to activate the less secure option before it could finally be chosen. Completely hiding an option from a user (as we did in version B) may introduce problems for some users that are confident of skipping the error message but are then unable to find this option.

5.9 The User Study Web Browser

This chapter is based on the work that was part of the bachelor thesis "Creating a Web Browser for User Studies on Security" by the student Alexander Gundermann [114].



Many of today's developed anti-phishing tools are browser based. The related work chapters 3.5 and 3.6 reported about some of those projects and other projects in this thesis also created browser user interfaces that had to be evaluated (see subchapters 5.2, 5.5, 5.6, 5.8). When evaluating such tools and interfaces in a user study researchers want to see how people react when the tool warns them about a potential phishing website and whether they are stopped from disclosing their private data – in other words whether the user intervention works.

5.9.1 Web Browsers Usage in Today's Experiments

Generally speaking one could simply present the user with real existing websites and monitor the users' behavior when visiting those. When doing this, a lot of parameters are important (e.g. not disclosing the fact that security is the main goal of the study). Many of those parameters have been discussed earlier (see section 3.7) and recommendations for their correct use will also be given later (see section 7.2). In this section we will present a solution for the problem that using existing websites for such a study is not as easy as it might look like.

Sending the user to non-malicious websites within a study might not denote a big problem in general, but as phishing tries to gather personal data the original websites that would be needed for such a study usually contain login or order forms. If the backend web server of such a study is not under the control of the researcher, it might get hard to carry out such a study, as a login with a username and password could fail and depending on the scenario of the study the user will need to perform actions within such a website (e.g. buying something). Making all this happening on real websites will sooner or later yield problems. Another possible problem would be a change of the website functionality or design while the study is in progress. Such an uncontrollable change may affect the experiment outcome. Summing this up, it would be best to have full control over the non-malicious websites.

For phishing websites the same problems exist and additionally it would definitely not be a good idea to have participants loose their credentials to real phishers within a research study context. Instead of design changes it is also most likely that a phishing site will go offline during the course of a running experiment.

To solve all those problems researchers have come up with various solutions for their experiments in the past:

- **Screenshots:** In some user studies the researchers tested their new concepts with screenshots of browsers rather than with an actual running browser. People are shown screenshots of different situations and have to reason about what they think and what actions they probably would perform next. Biddle et al. [29] did such a kind of study when evaluating a new kind of SSL certificate information window. A big advantage of the use of screenshots is that the websites and imagery used can be best controlled. In case it is needed to have a different URL than the one of the page actually photographed it is possible to simply merge different kinds of screenshots or introduce changes using a photo editing software. A big problem with such a kind of evaluation is the missing interactivity. The participants cannot interact with the browser as they normally would and actions they would possibly take, have to be described verbally instead of just recording what interactivity would really be performed.

- **Simulated Browser:** A more advanced option is to create a simulated browser. Wu et al. [324] created such a 'virtual browser in the browser'. They used HTML to generate a real looking browser within a browser window that users could interact with. Using this method they had full control over the browser interface and the content, whilst still having a certain degree of interactivity. Such a simulated browser was also part of our project evaluation in the previous subchapter 5.8. The downside of this method is that it will never get possible to model all different aspects of a browser within such a graphical fake browser and even if it was possible, it would still run inside another larger browser window finally distracting the user.
- **Registering Own Domains:** In case the experiment only needs domains that are not yet registered by other companies the researchers could simply register those domains and host their own (phishing) websites there. For legal reasons the access to those websites should still be restricted to experiment participators in a way (e.g. using an IP-whitelist). The problem with this method is that for most experiments original non-malicious websites should also be used which are already registered. Egelman et al. [78] used this approach and registered their own phishing domains.
- **Rerouting Contents on Network Level:** The most prominent idea used to display arbitrary websites within a real world browser, is by using network traffic diversion in various ways to make the browser load different content than it would normally load. One possibility is to use a proxy server that sits between the experimental browser and the target websites and diverts all traffic in both directions. A proxy server is generally able to modify the content of every request and result that passes it. Requests that should go to a non-malicious website can be intercepted and replaced by answers coming from another source instead – used by [259]. A problem with proxy servers is that they cannot alter the content of an encrypted connection without the browser noticing it. This means that for SSL connections this is not really an option. The same effect as with a proxy server can be achieved using the hosts file [304] of the experiment computer. This file can be used to divert HTTP requests to other IP destinations than they would usually go to in case of using the answer from a standard DNS request. A request to an existing domain could hence be diverted to a different computer answering this request. In this case the connection can even be encrypted with a self-signed

certificate to simulate an encrypted connection. This approach has been used in our project described in subchapter 5.5 and by other researchers (e.g. Dhamija et al. [67]). The only downsides of this approach are that the effort for changing all the necessary settings is relatively high but even worse is still the fact that it is impossible to fake extended validation certificates and that the certificate info will always show that the certificate is not the same as the one from the original website.

- **Building an Own Browser:** In some cases researchers built a whole web browser for their experiment or modified the code of existing web browsers to change the behavior in the way they needed it. Sobey [269] modified the source code of the open source Firefox browser to integrate custom designs in different own builds whilst Sunshine et al. [280] replaced the warning screens of the Internet Explorer browser which are stored in a library of the browser. Even though these attempts are a lot of work, they provide the most original interactivity whilst fully integrating the experiment aspects that are investigated. These approaches are also similar to what we will present in the current chapter.

5.9.2 Universal Browser Manipulation

As one can see from the previous examples, diverting website traffic and faking security information on a network level is very hard and sometimes even impossible. This is by design as network security and certificates should be able to guarantee that the traffic originated from an intended sender. We argue that such manipulations can be done much more easily on the software more specifically the user interface level.

For the purposes of future user studies and as a proof-of-concept we tried to build a browser extension that can be used to make any browser security or connection indicator look like as if website traffic originated from any other given source. We called this developed plugin “certificate faker extension”.

The three main goals when building the plugin where:

- **URL Spoofing:** With a URL spoofing component it is possible to make any website look like as if it has been loaded from any other given source. This can be used to make websites loaded from a lab web server look like as if an original website has been loaded or it can even be used to create perfect phishing websites by changing the URL of an original website to something that looks like a phishing website.
- **Certificate Spoofing:** Making the security indicators of a browser look like as if they have a certain status was the biggest problem of the methods used by researchers so far. For experiments it should possible make the security indicators look like if any arbitrary certificate is loaded and the indicators should all represent the respective security status (e.g. an EV-SSL certificate has been validated).

- **Customizability:** Finally it was of importance to be able to customize all these options easily without having to recompile the browser for every change or even to support a large number of different websites throughout a longer user study.

5.9.3 Developing the Extension

To achieve our goal of such a browser modification we saw three different possibilities with different advantages and disadvantages:

- **Recompiling the Browser:** Since the Firefox browser is open source and as previous researchers had also had success for their research by recompiling the browser with extensions to suite their needs [269] we considered this option first. The trusted certificate authorities for example are precompiled within the browser source code and can only be changed by recompiling it completely. In fact, recompiling such a large piece of software is a tough piece of work and each modification to the code would have to be reapplied with each new version of the browser that would be released. Although such a modification would have worked out, we found out that there was no need at all to replace the certificate authorities since we were only interested in manipulating the visual appearance of the security indicators.

- **Changing JavaScript Browser Components:** The whole visual appearance of the Firefox browser is produced by a JavaScript based frontend. This code is not compiled into the executable at compile time but is instead loaded from an additional file called “omni.jar” at runtime. This archive can simply be opened and altered and if the files are extracted into the root directory of the browser they are even used in the uncompressed state. The problem here still is that modified files in this archive would be changing in future version of the browser and hence the changes would also have to be reapplied with each new browser version.

- **Building a Firefox Extension:** Firefox extensions are plugins that can be installed into the Firefox browser and add certain functionality to the browser that can be beyond what standard JavaScript of a website is allowed to execute. An API documentation[22] of Mozilla reports the functions that can be used to access for example cookies or the stored history or even write to files or databases. After some tests we found out that the JavaScript code of Mozilla and the JavaScript code of extensions are executed with the exact same security privileges in the exact same context sharing all variables and namespaces. As extensions are loaded after the main browser components have been loaded it gets even possible to shadow[23] existing functions and replace them with

[22] `https://developer.mozilla.org/`

[23] In programming, variables or functions can shadow previously created definitions of the same name [313]. In our case we shadowed existing functions of the browser such that the browsers own routines would call our new methods instead.

own code. This is also true for functions related to the visual appearance of the security indicators. With those findings it gets easy to build a Firefox extensions that would change methods in all internal JavaScript functions of the "omni.jar" wherever needed. This will work for all future Firefox versions as long as no changes to the execution policy of extensions or changes to the architecture of the respective functions would happen.

Implementation

The most important handler for our plugin was to detect every change to a website that was currently loaded in any of the windows or tabs of the browser. Whenever the website or the URL changes, the visual indicators displaying the URL or the security indicators were then adapted if necessary. Where possible we used officially documented listeners and otherwise simply created our own listeners by overriding the native Firefox functions and calling the overshadowed parent function with adapted parameters.

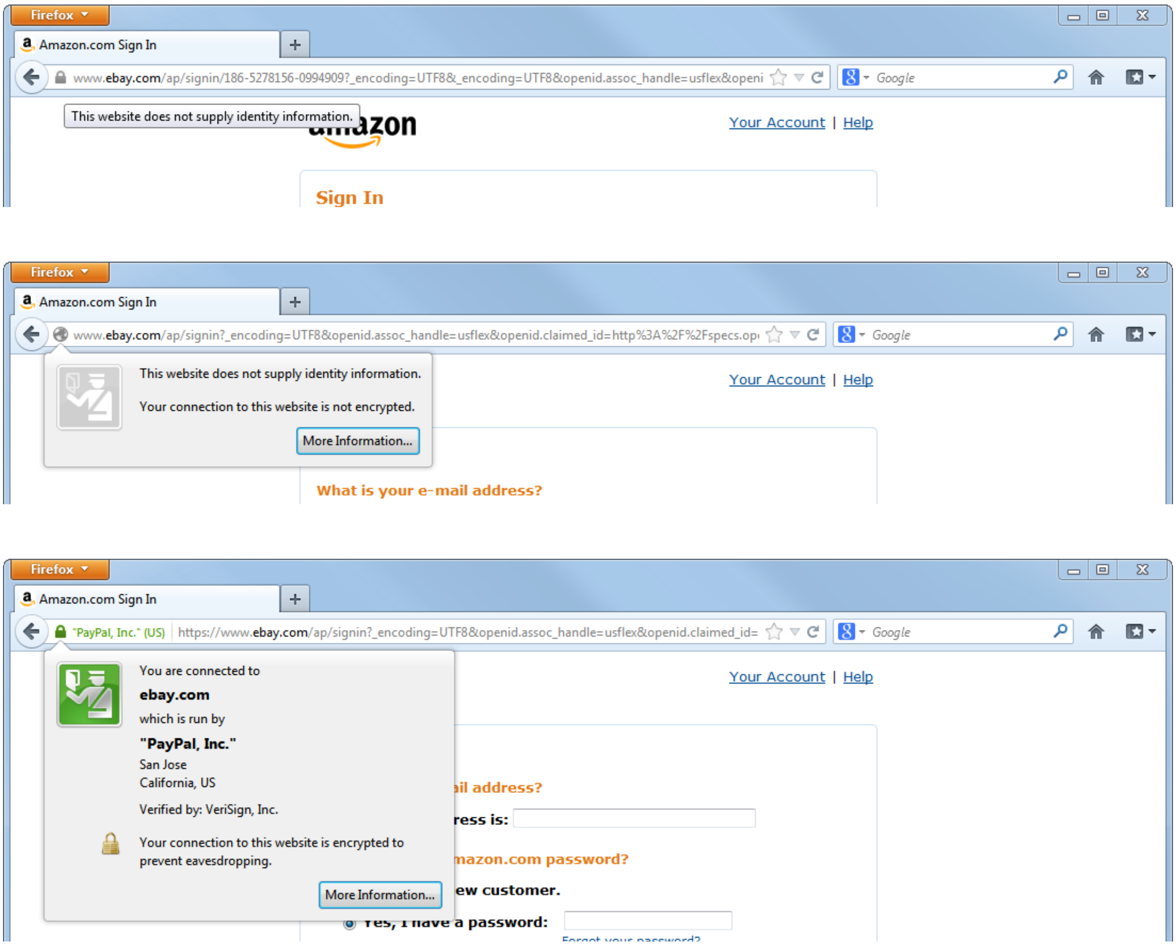
Spoofed Indicators

Throughout the browser we spoofed a lot of different security and location indicators to make sure that the browsing experience is completely spoofed. In the address bar these have been aside from the **URL**, the color, the label and the status of the **site identity icon**. The site identity icon is the colored area next to the URL bar that changes its size, text and color whenever the user is connected to a server using a secure connection. In addition to that we also had to spoof the **mouse-over** text that is displayed when the user hovers over this area (see figure 5.45 for the spoofed indicators). When clicking the site identity button the user sees an **identity popup window** that provides more information about the encryption of the connection (see figure 5.45). A "more information" button within this dialog brings up a detailed **dialog with properties of the security certificate**. This window and all its properties had to be spoofed as well (see figure 5.46). Besides all this SSL related information the website URLs had to be replaced in all other places. Whenever the user hovers a link in a website the target URL is displayed in the lower left corner of the browser. This **link target indicator** also had to be spoofed as well as all links in the **HTML source code window**.

Controlling the Spoofing

Controlling the plugin was either possible by a user interface (see figure 5.47) or by JavaScript code (e.g. from another plugin). In both cases the user can specify two different types of configuration components to setup the plugin:

- **Certificate entries:** The user can configure arbitrary virtual certificates (without having them to be actually trusted by any certificate authority). In the configuration dialog



www.ebay.com/ap/signin/186-5278156-0994909?_encoding=UTF8&_encoding=UTF8&openid.assoc_handle=usflex&openi

This website does not supply identity information.

Sign In

www.ebay.com/ap/signin?_encoding=UTF8&openid.assoc_handle=usflex&openid.claimed_id=http%3A%2F%2Fspecs.op

This website does not supply identity information.

Your connection to this website is not encrypted.

What is your e-mail address?

"PayPal, Inc." (US) https://www.ebay.com/ap/signin?_encoding=UTF8&openid.assoc_handle=usflex&openid.claimed_id=

You are connected to
ebay.com
which is run by
"PayPal, Inc."
San Jose
California, US

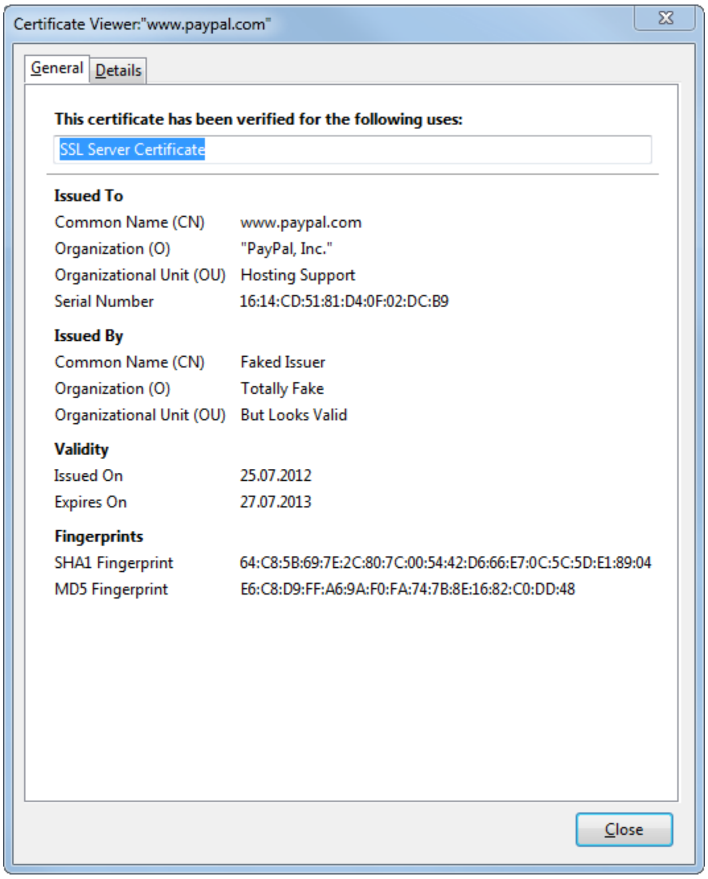
Verified by: VeriSign, Inc.

Your connection to this website is encrypted to prevent eavesdropping.

Figure 5.45: Three different examples for the spoofing done with the certificate faker extension. For these screenshots the loaded website is the usually SSL secured login website of amazon.com. The URL in this case is rewritten to look like ebay.com and the certificate status is set to either insecure (with mouse over [top]; popup identity dialog [center]) and extended validation [bottom].

the certificate status can be easily selected ranging from the standard “insecure” connection over the standard SSL-DV certificate and the extended validation (EV) certificate. The error states “security exception” and “mixed content” can also be used. By setting other virtual certificates that have been previously created as a parent. It is possible to construct a whole virtual certificate chain.

- **Website entries:** Using regular or wildcard expressions the user can define URLs that are matched and afterwards replaced by a target URL. The part of any URL that is matched by the expression is then replaced by the new URL part. Setting the expression to “ebay.com” and the target URL to “amazon.com” would result in eBay pages looking like as if the were loaded from “amazon.com”. For each of those entries it is also possible to set a fake certificate that overrides the certificate of the actual con-



This certificate has been verified for the following uses:

SSL Server Certificate

Issued To

Common Name (CN)	www.paypal.com
Organization (O)	"PayPal, Inc."
Organizational Unit (OU)	Hosting Support
Serial Number	16:14:CD:51:81:D4:0F:02:DC:B9

Issued By

Common Name (CN)	Faked Issuer
Organization (O)	Totally Fake
Organizational Unit (OU)	But Looks Valid

Validity

Issued On	25.07.2012
Expires On	27.07.2013

Fingerprints

SHA1 Fingerprint	64:C8:5B:69:7E:2C:80:7C:00:54:42:D6:66:E7:0C:5C:5D:E1:89:04
MD5 Fingerprint	E6:C8:D9:FF:A6:9A:F0:FA:74:7B:8E:16:82:C0:DD:48

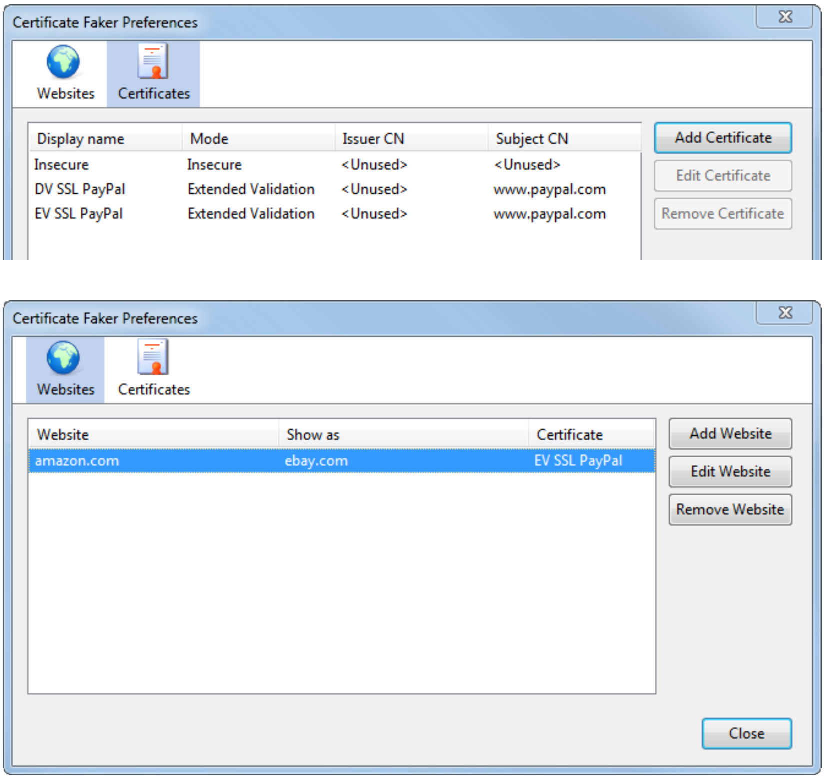
Figure 5.46: This dialog shows a spoofed certificate dialog window with a spoofed PayPal certificate. The issuer fields have been filled with other text.

nection. As the replacement of the URLs is only done on the user interface level the browser won't report any errors that the certificate URL does not match the loaded URL.

5.9.4 User Study: Validating the Extension

To test whether our modifications did work out and whether users would be able to recognize the spoofed indicators in any way, we conducted a user study testing whether phishing websites can now be shown as perfectly believable original websites (type 4) and whether authentic websites can now be disguised as phishing websites (type 3). As control conditions we additionally had unmodified original (type 1) and phishing websites (type 2) (see table 5.17).

Besides the type of fake we used four different websites for our tests that were overall well known and also popular phishing targets: paypal.com (A), facebook.com (B), ebay.de (C)



Certificate Faker Preferences

Display name	Mode	Issuer CN	Subject CN
Insecure	Insecure	<Unused>	<Unused>
DV SSL PayPal	Extended Validation	<Unused>	www.paypal.com
EV SSL PayPal	Extended Validation	<Unused>	www.paypal.com

Certificate Faker Preferences

Website	Show as	Certificate
amazon.com	ebay.com	EV SSL PayPal

Figure 5.47: The certificate faker user interface allows to create fake virtual certificates entries that can afterwards be assigned to arbitrary websites. For each website URL rewriting can also be configured.

		Input	
		Authentic	Phishing
Modi-fication	Original	1 Authentic website	2 Phishing website
	Reversed	3 Authentic website that seems to be phishing	4 Phishing Website that seems to be authentic

Table 5.17: The four different conditions of spoofed and unspoofed, authentic and phishing websites used for our within-subjects experiment.

		Original	Phishing
Websites	A	paypal.com	pajpal.de
	B	facebook.com	facebqpk.com
	C	ebay.de	eblay.de
	D	amazon.de	amazqn.de

Table 5.18: The domain names and corresponding phishing domain names used for the user study.

and amazon.de (D). We assigned a possible phishing domain to each of the websites (see table 5.18).

Study Methodology

We did a within-subject study with eight participants. Each participant saw each type of website modification (1 to 4) combined with one of the websites (A to D). The design was counterbalanced using two Latin squares that were combined and swapped for the second half of the participants.

The study was conducted in the lab on a standard computer running Windows 7 and Firefox 8. Since we didn't want to confront people with live and online phishing websites (type 2) we created phishing websites on a local server and used the DHCP server of the local network to divert the traffic to our local servers. This did not affect plugin testing.

We configured our plugin to not intercept the user interface for any websites in the original conditions (type 1, type 2) but removed any encryption indicators when displaying original websites as phishing (type 3) (as phishing websites are usually not encrypted). For phishing websites we wanted to make look like the original we replicated the complete certificate chain the original website has with our plugin (type 4).

Since we wanted to find out whether the most security aware user would be able to find our modifications to the browser we primed our participants for security by telling them that this study was about phishing. What we didn't disclose to them in the beginning was the fact that we had modified the browser with our plugin.

After signing a consent form and filling out a demographics questionnaire we calibrated the eye tracker and let the participants examine each website. In between the websites we wanted them to rate whether the website was phishing or not and how confident they were with their decision (5-point Likert scale). They were allowed to interact with the browser in any way they wanted.

Study Results

Three of our eight participants were female with an average age of 26 years (19 to 31). All but one participant used Mozilla Firefox as their primary browser. On average the participants spent 6 hours on the Internet per day. All of our participants could explain the term phishing but had never fallen victim to a phishing attack.

On average it took participants about 160 seconds to judge each website they were presented with. Phishing websites were judged faster (avg. 95 seconds). Table 5.19 shows an overview over the judgments that participants gave. Looking at the results one can see that for all but two ratings our modified browser was able to fool people into thinking phishing sites were originals or vice versa. In case of the phishing sites being manipulated to look like the originals two people correctly decided that this website was phishing. But looking at the assessments of the original unmodified website one can see that people also rated two websites as being phishing here. We think that having primed our subjects for security and to look out for phishing pages, they were in general more likely to rate websites as being phishing. Two of the 32 decisions made had to be disqualified: as people were allowed to freely use the browser two of them accidentally changed the website they had been on navigating away from the phishing website to the original website. They then judged this website instead of the one we had initially brought them to.

Concluding one can see that besides the random noise, people were not at all able to tell the spoofed websites apart from the unmodified ones.

	Participant							
	1	2	3	4	5	6	7	8
1 Real authentic website	🟩 PayPal	🟩 amazon	🟥 facebook	🟥 eBay	🟩 PayPal	🟩 amazon	🟩 facebook	🟩 eBay
4 Phishing faked to look authentic	🟥 amazon	🟥 PayPal	🟩 eBay	🟥 facebook	🟩 amazon	🟥 PayPal	🟥 eBay	🟥 facebook
2 Phishing Website	🟩 facebook	🟩 eBay	🟩 PayPal	🟩 amazon	🟩 facebook	⬜ eBay	🟩 PayPal	🟩 amazon
3 Authentic website faked to look like phishing	🟥 eBay	🟥 facebook	⬜ amazon	🟥 PayPal	🟥 eBay	🟥 facebook	🟥 amazon	🟥 PayPal

🟩 Correct Decision 🟥 Wrong Decision ⬜ Disqualified

Table 5.19: Decisions of the eight participants of the user study for the certificate faker extension. In most cases participants answered correctly for the websites that have not been modified and incorrectly in cases that had been modified by the extension.

5.9.5 Research Results

Although we were not able to use this study browser throughout the work that has been done within prior projects of this thesis, this work clearly shows that there is a simple way of conducting authentically looking phishing studies without having to compile ones own web browser.

IM *How can User Intervention be Measured?*

Whenever building new methods of user intervention in the phishing domain it is important to validate the work with a real user base. This means that users have to be confronted somehow with potential malicious websites to see whether the intervention mechanisms work and with original websites to see whether eventual errors in the intervention mechanisms can still be detected by the user.

Over the last years researchers have come up with a wealth of mechanisms to create such study situations (presented in section 5.9.1): Ranging from screenshots instead of real browsers to completely re-engineered open source systems they all have their downsides by either missing interactivity or validity. Although applicable in some situations, we showed a more general solution in this chapter that is even easier to implement.

The core idea to this type of browser plugin is not to try to evade the network security mechanisms on the network security level. As they are designed to withstand most attacks it is close to impossible to successfully fake something like an extended validation certificate. Since the user interface of the browser only shows a graphical representation of the network traffic that is going on, it is much easier to change the visual indicators without affecting the network traffic. We showed that this type of modification works well for all different kinds of security indicators, hence allowing for most accurate experiment setups.

Take Home Messages

- **5.1 Phishing Website Test Set:** Phishing test sets are vital to understand the landscape of possible attacks and to test phishing detectors. A phishing test set should cover many of the possible input variables a detector could make use of to make it applicable to as many detectors as possible. The most important of these are URL, HTML content, screenshots, domain info and linked content. Having a standardized test set structure can make test sets interchangeable. The complete coverage of the phishing landscape is another important point of a good test set: too small test sets might suffer of grouping effects and tests missing original websites are not suitable for testing false positives.

- **5.2 SecurityGuard Website Status Rollup:** To find and evaluate new user intervention concepts it is not always necessary to build a huge new kind of detector. Starting off with a user centered design process by gathering ways that the users might want to assess website security can yield new forms of user intervention and finally even new forms of detectors. A rudimentary implementation of such a detector can already help to gain a lot of insights for both the user intervention mechanism and the detector. Besides this, qualitative evaluation methods and field studies are very good means to assess certain problems a specific concept might have, but in the end it does not provide any insights about hard performance facts of the detection method or the user intervention mechanism. Such measurements should always be done in combination with another type of study.

- **5.3 Community-based Rating Intervention:** Although the idea of community-based website ratings for security and trust assessment of websites is already in productive use we were not able to find large effects that our plugin score or even appearing warnings were able to change or influence the participants attitude towards the websites. This certainly does not have to mean that the concept does not work in general. It could have also been due to the user intervention feedback in form of a small indicator that we used. For future user interventions based on this a much more visible feedback mechanism would be needed, especially in the case of critical ratings. Other important findings here were that one simple value for such a security recommendation is not enough. The users want to see more reasoning for the given values. Who was it who gave this rating and why? Individual comments might help in such cases. Another example would be to include a kind of testimonial for each website written by someone else.

- **5.4 Spell Checking to Detect Fraudulent Websites:** URLs are a type of Internet data that allow to detect phishing attempts to some extent. As many phishers apply specific modifications to their domains to make them look more trustworthy this can be used to detect those URLs that try to be similar to original URLs. A detection rate of about 51% as achieved within this project is by far not enough for a stand-alone detector and hence such a concept would need to be enhanced or coupled with another detector. Interestingly, the quality of a phishing attack seems to play an important role for the evaluation of a phishing detector as some detectors might be able to detect more attacks of higher quality (as in our case).

- ➥ **5.5 Data Type Based Security Dialogs:** The involved data types within a website already tell a lot about whether a website can at all be an attack. Using such context data makes it possible to ease detection a lot and bring down false positives. The number of websites involving critical data is much more limited than the overall number of websites that a user visits. Together with the concept of semi-blocking dialogs well reasoned warnings can be brought up at the right moment in time at the right place. However using the concept only with the data type filter and an empty whitelist, results in too many false positives in the beginning that need to be reduced by other means.

- ➥ **5.6 Enhancing SSL Awareness in Web Browsers:** Non-blocking indicators can influence users but a large area for the user feedback is needed such that the changes of the indicator can be easily noticed. Since this feedback area is not needed for user interaction the space of other interaction elements can be reused. We used background images in the browser (Personas) in case of this project. Rolling the concept out as a security plugin showed that more security professionals manage or dare to download new security extensions than novice users. With such a kind of deployment it is hard for new technologies and prototypes to reach the target audience of security novices.

- ➥ **5.7 Diminishing Visual Brand Trust:** Getting the users' focus towards non-blocking indicators is hard but can be done from the outside of their focus attracting the users' attention from there. Diminishing the users' trust to the content area of the browser where the actual focus is, did not bring any security improvements in our test. Altering the browser contents by changing logos and text did have an effect of confusing participants but this confusion did not lead to more security attention.

- ➥ **5.8 Visual Image Comparison For Phishing Detection and Reporting:** Visual similarity as means for phishing detection already is able to detect a large number of phishing websites by comparing the user's current screen contents against a list of known attacks and original websites targeted. However, this part of the protection process still needs some future enhancements to end up at detection rates that are desirable. On the user intervention side this concept seems to be the ne plus ultra as participants can easily understand what is going on and generate mature decisions out of the warnings, not only leaving malicious websites but also recognizing detector errors to a large extent.

- ➥ **5.9 The User Study Web Browser:** When testing user intervention methods in the browser the way the user is presented with the new intervention concept is vital. Screenshots of browser windows lack interactivity and do not allow the user to explore the concept to the fullest. Real web browsers were used in certain experiments so far but usually with the downside that changes to the network traffic flow and the security can often be detected by the participant. Using modifications to the visual notifications of the browser it is possible to create indistinguishable spoofed websites that can then be used for any type of intervention experiment.

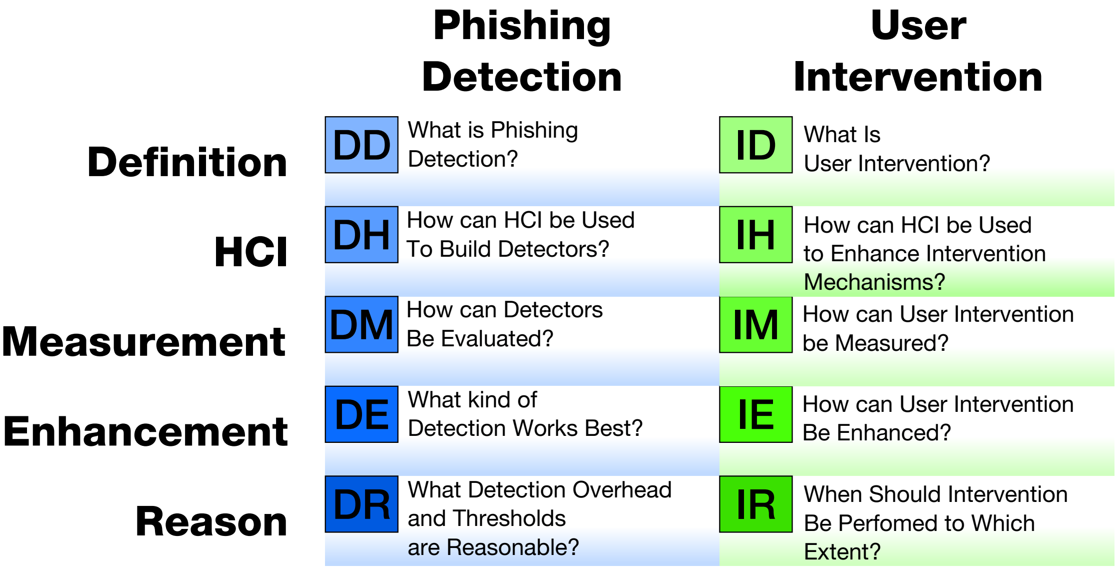
Chapter 6

Aggregated Results and Derived Recommendations

During each of the nine subchapters in chapter 5, I already gave project specific answers towards each research question wherever applicable. Within this section I want to summarize those findings again and paint the big picture of the answers to all the research questions. For some of the research questions these findings lead to derived recommendations. Afterwards I will take a brief look on how the results taken in the light of detecting phishing attacks, can be transferred to more general problems of usable security. Finally, I will highlight the important interaction that exists between detector and user intervention development and the different stakeholders, presenting a new model of these dependencies in the last section of this chapter.

6.1 Answers to the Research Questions

Figure 4.2 in chapter 4 gave an overview over the different research questions of this thesis on both dimensions – phishing detection and user intervention. For convenience it is again included within this chapter (see figure 6.1). The research questions of this thesis tried to bridge the two often separated worlds within computer research of finding security solutions and creating usable interface concepts – the user intervention. To bridge these two worlds using an HCI perspective I looked at them on five different levels (definition, HCI, measurement, enhancement and reason) and in the following I want to summarize the research findings for each single one of those questions. I want to invite the reader to have a short look back to chapter 4 to go over the details of these research questions again. I will reference the involved projects by subchapter numbers (see figure 6.2 for a quick reference).



	Phishing Detection		User Intervention	
Definition	DD	What is Phishing Detection?	ID	What Is User Intervention?
HCI	DH	How can HCI be Used To Build Detectors?	IH	How can HCI be Used to Enhance Intervention Mechanisms?
Measurement	DM	How can Detectors Be Evaluated?	IM	How can User Intervention be Measured?
Enhancement	DE	What kind of Detection Works Best?	IE	How can User Intervention Be Enhanced?
Reason	DR	What Detection Overhead and Thresholds are Reasonable?	IR	When Should Intervention Be Perfomed to Which Extent?

Figure 6.1: Overview of the ten main research questions of this thesis being split up in two dimensions on five different levels (duplicate of figure 4.2).

6.1.1 Phishing Detection

Phishing is not like many other computer security threats. It has a social engineering component. This creates new problems but also offers new possibilities when creating detectors. This section will provide answers towards the five research questions of the "phishing detection" dimension.

DD *What is Phishing Detection?*

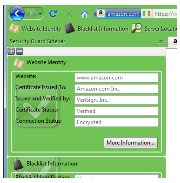
In section 2.1 I already gave a lot of examples for the definition of phishing and included my own definition of phishing for the course of this thesis. I understood phishing detection within this thesis as the ability of a certain piece of software to automatically discriminate given websites between being a phishing attack or being a non-malicious website.

In project 5.4 for example this piece of software was a detector that classified URLs into potentially phishing or not. The piece of software hence solves a binary classification problem with using input parameters – here a URL. In project 5.8 the input towards the software were screenshots of websites while project 5.2 used domain specific parameters. Detectors presented in the related work chapter 3 used other means as an input. In sum these all are attributes of a given website. A second component that might be used by a detector is the user's context – if known. In project 5.5 we used it as a filter for possible websites. Taking all this into account phishing detection can be understood as a function mapping website attributes and user context to a phishing result:

5.1 Phishing Website Test Set

Having an extensive phishing website test base is important for getting valid results for detector testing throughout the research. This chapter introduces into the important aspects of such a pishing test set and reports the process of a test set that has been built up during this thesis.

DD	ID
DH	IH
DM	IM
DE	IE
DR	IR



Website Identity

Website: www.amazon.com
Certificate Issued To: Amazon.com Inc.
Issued and Verified by: VeriSign, Inc.
Certificate Status: Verified
Connection Status: Encrypted

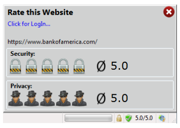
More Information...

Blacklist Information

5.2 SecurityGuard Website Status Rollup

What technical properties of websites are interesting for users and how can they be offered towards them within a user intervention mechanism? Within this project we built a rating and reporting systems that displayed technical data concerning the current website within a status bar in the browser.

DD	ID
DH	**IH**
DM	**IM**
DE	**IE**
DR	IR

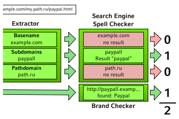


Rate this Website
Click for Login...
https://www.bankofamerica.com/
Security: Ø 5.0
Privacy: Ø 5.0

5.3 Community-based Rating Intervention

In real life people often ask others for security and privacy advice even about Internet websites. Can such a concept be used online and is it more attractive to users as they make use of it offline? Within this project we built a user intervention method as a browser plugin and evaluated possible effects.

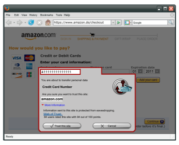
DH	ID
DH	**IH**
DM	**IM**
DE	**IE**
DR	**IR**



5.4 Spell Checking to Detect Fraudulent Websites

URLs are an important indicator for detecting phishing attacks as they cannot be as easily impersonated. Can similar looking URLs be used to detect phishing attacks automatically? In this chapter we present a detector and its evaluation that is based on URL similarity.

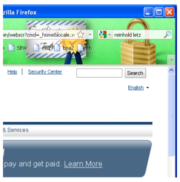
DD	ID
DH	IH
DM	IM
DE	IE
DR	IR



5.5 Data Type Based Security Dialogs

In this subchapter we take the advantage of the fact that phishing only happens if critical types of data (e.g. credit card numbers) are involved. This can be used for filtering incoming attacks and allows to create a user intervention method that takes the users' context into account.

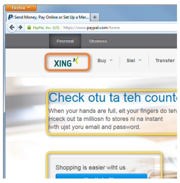
DD	ID
DH	**IH**
DM	**IM**
DE	**IE**
DR	**IR**



5.6 Enhancing SSL Awareness in Web Browsers

Non-blocking indicators are usually said to remain unnoticed by the users. This has been proven for lock icons and other smaller security indicators. Within this project we test this again by using the whole background of the browser user interface to report the SSL status.

DH	ID
DH	**IH**
DM	**IM**
DE	**IE**
DR	**IR**



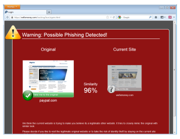
Check otu ta teh count

Shopping is easier wiht us

5.7 Diminishing Visual Brand Trust

The content area of the web browser is the most important place for users to assess the trustworthiness of a website although it is easily impersonated. In case of this project we looked at whether small website content changes lead to a stronger focus on other security indicators in the browser.

DH	**ID**
DH	**IH**
DM	**IM**
DE	**IE**
DR	IR

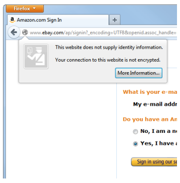


Warning: Possible Phishing Detected!

5.8 Visual Image Comparison For Phishing Detection

The visual similarity of impersonating websites towards their original websites makes users often trust into these faked websites. We built a detector for phishing websites using visual similarity and tested different image features as well as a user intervention method for this kind of detection.

DD	ID
DH	**IH**
DM	**IM**
DE	IE
DR	IR



Amazon.com Sign In

This website does not supply identity information.

Your connection to this website is not encrypted.

More Information...

5.9 The User Study Web Browser

Using live original and phishing websites in user studies brings along a variety of problems in a study setup that needs to fulfill certain parameters. Within this project we built a browser plugin that makes it possible to mimic arbitrary security situations and finally tested whether those changes can be detect.

DH	ID
DH	IH
DM	**IM**
DE	IE
DR	IR

Figure 6.2: Overview of all projects carried out throughout this thesis each showing which research questions have been covered (duplicate of figure 4.3).

$$f\,(\text{attributes(website)}, \text{context info}) = \begin{cases} 1 & \text{if is phishing} \\ 0 & \text{if is NOT phishing} \end{cases}$$

$$\text{attributes(website)} = \begin{pmatrix} \text{URL} \\ \text{HTML content} \\ \text{screenshots} \\ \text{domain info} \\ \text{linked content} \\ \dots \end{pmatrix}$$

While we had this strong binary classification in project 5.4 the detector in project 5.8 was able to return intermediate values of the website similarity. In case of such a detector outcome a threshold value has to be defined that converts the results back to a binary result.

In many cases I was able to use the intermediate results of the detection process to explain the binary detection result to the user (e.g. the different module results in project 5.2). Another example for such an additional detection result can be a suggested original website (for a detected phishing website). The detector that we developed in section 5.8 delivered possible matching original websites that made it possible to present a safe escape path to the user. I hence argue that the more intermediate values and details can be delivered by a detector the easier it gets on the user intervention side to transform this data into working and meaningful user intervention methods. Detectors that are built should offer access to intermediate values generated.

Pre- and postfiltering of inputs and outputs of a detector may also be part of the phishing detection process. In project 5.5 we filtered the website inputs only based on type of data that was entered on the website and used a white-list as a second way of filtering. Although no other means of detection were used the number of appearing warning messages quickly reduced to 4.5% without other means of detection.

I think that these different aspects of phishing detection present the most important aspects of what phishing detection is about. The final compiled model in subchapter 6.3 takes these aspects into account.

DH *How can HCI be Used to Build Detectors?*

Taking the definition from the research question above into account no direct human interaction with the detector is part of the detection process. However, I argue that the HCI should be an integral part of detector development as usability observation can yield to better detectors in two different ways:

In project 5.4 we developed our URL-based detector looking at the way the human attackers craft their URLs by introducing typos and placing original URLs and brand names within the URLs of their phishing websites a detection style that seems especially suitable for high quality attacks. Project 5.8 made use of the observation that phishers most often try to closely mimic the look and feel of original websites. Looking at the behavior of attackers and how they set up their attacks can hence be very useful to finding new kinds of counterattacks.

The second use of HCI in detector development is based on the fact that each detection result in the end has to be presented to the user to take a final decision. We deliberately extended many of our detectors within this thesis to yield results that can be used for the user intervention method (cf. research question DD). The best results were achieved in project 5.8 where we used similarity scores, tentative original websites and website screenshots that would be delivered from a detector and achieved 97% of true positive and even 49% of true negative decisions for our warning design A. In project 5.2 we found the subdetector values to display by interviewing users prior to creating the detector.

I recommend to take human properties of users and attackers into account the moment a detector is developed. Incorporating the human properties already at the stage of building a detector can greatly help to achieve model similarity between the implementation and the conceptual model [54] and make the detector results more understandable towards the user. Possible points of action to detect phishing are all the aspects of Internet communication in general and most of them have been already taken into account by either research or practical development. Within this thesis I often used properties that are very close to the actual user interaction. No matter which properties are used it might make sense to take a look at what the users think. Which properties actually are best for the detection and finally the user intervention process will be subject to future research.

DM *How can Detectors Be Evaluated?*

Throughout this thesis different detectors have been built (projects 5.2, 5.4, 5.8). A general measurement for the quality of such a detector is to use true and false positives and negatives. Depending on the kind of detector different types of inputs towards the detector are necessary. In project 5.8 we used screenshots of the websites as an input while in project 5.4 the URLs were the only necessary input variable towards our detector. The detector in project 5.2 took mostly technical properties (e.g. traffic encryption, domain age) but also some user based properties into account (e.g. number of prior visits). This shows that depending on the detector a lot of different input variables are necessary.

For a lab testing of a detector hence a stable set of those inputs is needed (e.g. to test a different detector over and over again). Testing with live data will mostly be problematic as websites and especially phishing attacks change quickly which would result in different testing results independent of the changes to the detector. In case a test set covers input types for multiple detectors (as we did in project 5.1) it can be even used for direct comparison between two different types of detectors.

I also found that although detector testing is very technical, qualitative feedback of users can also be a property for detector evaluation. In combination with the test of the user intervention method in project 5.2 the users mentioned that the location data was one of the most important types of information for them. Location information was not used by our detector within this project but according to these results it possibly should have been used. The target group of a detector result in a way is always the user, which is why the users' expectations towards a detection process play an important role within the detection process.

The "quality" of an attack also has an influence on the detection result (as we saw in project 5.4). In this case we had experts rate the "visual quality" of phishing attacks and found different detection results for the different quality groups of our detector. This shows a tendency towards attack quality being an important factor to include in detector evaluations. However, future studies will be needed to show whether the quality of an attack really has an impact on how well people fall for the attacks and how "phishing quality" can exactly be defined.

In project 5.8 we also used the FP and FN equilibrium threshold value as a measure for detector quality. I propose to use this value as means for detector comparison when reporting detector results in scientific articles as it combines information from both FP and FN performance in one value. Whether this value is a well working measure of detector quality will need to be tested in further studies to see whether it reflects detector quality well enough.

From the experiences and findings within this thesis I compiled a list of recommendations of how a test set should optimally look like – most of which I tried to fulfill in project 5.1:

1. A test set should always test the detector not only for true positives and false negatives – how well it detects phishing attacks – but also for its behavior with original websites – false positives and true negatives.

2. The number of test entries within the test set should also be rather high (I recommend several thousand test entries) as collecting phishing websites over shorter periods of time may introduce bias towards certain brands (see project 5.1).

3. Test set data should be collected as snapshots of the state of each website as the websites and especially the availability of phishing websites change within minutes. Having a snapshot can guarantee that the same test results can be reproduced during different executions of the testing process.

4. The test set should also be as unbiased as possible towards a detector. It should not contain test sites that are specifically suited to pass the tests nor should it contain more impossible test cases than would exist in a real world setting.

5. As the phishing landscape is always changing, it should be as up-to-date as possible not using test data that is several years old.

6. If possible the data that is collected for a test set should contain more parameters than needed by the intended detector to enable the testing of other detectors with the same test data later on.

7. Linking each website to a respective brand name where possible extends the whole testing framework to make it even possible to judge how well a detector is able to

detect a matching brand. Using this in project 5.8 made it possible to find out how many brands were correctly identified by which detector.

8. Taking quality ratings for the different websites makes it possible to see whether a difference exists in the detector performance for high-quality or low-quality attacks. Our URL detector in project 5.4 worked better for high quality websites. The definition of quality in this case is a highly debatable topic. As the users are the critical target group the quality ratings should resemble somehow the likeliness of falling for a phish. However, the likeliness of falling for a specific phish is not easily determined. The most important aspect for a user to judge the authenticity of a website is the visual content. Since visual similarity is a more graspable concept I argue it is the best match for determining phishing attack quality.

Some of these aspects are to a small extent contradictory: having an up-to-date test set on the one hand contradicts with the principle of reusing test sets for similar tests. To solve these problems in future tests it would be best to introduce some kind of standard for test sets that specifies which data needs to be collected for a test set and how it can be accessed by testing frameworks. Taking this idea further, a standard test set acquisition and execution framework would be best. The framework and its test set snapshots could be easily shared and compared between researchers and their different kinds of detectors.

A test set to stand up towards the testing needs of most detectors that have been presented so far would need to include a broad range of different properties. Besides the URLs of the test websites and their classification as phishing or non-phishing, stored HTML contents are important for many detectors. Complete website snapshots with a certain link depth should be available to contain all linked files like CSS information – information about the styling of the website –, JavaScript, images and the sub pages of a URL. Technical details about the domain should also be available (e.g. when was the domain registered?, who is the owner?, reverse IP-lookups, server locations, the SSL certificate any many more). Different screenshots of the respective website content, the browser or the overall rendered page – possibly even using different types of browsers – can make it possible to evaluate detectors based on the visual appearance. Finally quality scores and brand identification could be assigned to all test data where possible to enable the aspects seven and eight presented above.

DE *What Kind of Detection Works Best?*

Throughout related work and this thesis many kinds of detectors have been tested using different input parameters.

In our project 5.2 we used mainly **technical properties** of the connection or domain (e.g. the time a domain is registered, country, SSL status). Although we displayed partial results for smaller amounts of data to our users I argue that these kinds of detectors will only work if they combine several attributes. In our project we identified those types of data that seemed interesting to the users and determined how they would like to see them in a user intervention method. Garera et al. [104] (see section 3.5.2) used such features with a stronger focus towards the detector.

I see the **URL** as a special kind of technical property as the user is more in touch with – seeing or typing it in the address bar. Even only looking at spelling mistakes and brand names within URLs we were already able to mark more than 50% of the phishing websites as suspicious in our project 5.4. Being small pieces of data, URLs can be quickly handled by a detector using millions of them in milliseconds. Even though we used a very complicated and slow approach sending different portions of them one by one to a server not traditionally suited for this kind of detection, our requests could have been easily optimized when building a professional detector instead of a prototype.

In related work **HTML content** is often used for detection. Due to phishing page polymorphism [158] I chose to use final rendered images that are generated out of the HTML code for evaluation in project 5.8. Within the project we achieved good results finding websites of the same brand in 92% of the cases. These results are not yet perfect but given the success of the user intervention methods for such a detector I argue that further improvement of such a concept will lead to a well working final detector.

To conclude the answer to this research question I would like to discuss the optimization possibilities of the different detector types but also the way attackers could possibly react. To find the optimum detector using technical properties it will be necessary to try out many different technical parameters and find the best way to aggregate them towards a final detection result in the future. Perhaps machine learning algorithms are best suited to find optimal thresholds and adapt to changes of the attacks over time. Once the quantifiers for the different parameters have been found, phishers still have the possibility to adapt to the detector mechanism by changing their attacks at the parameters having the biggest weight, usually without the user noticing. This is why I would not recommend using technical properties as the only means of detection. URLs are already more user-oriented technical properties which is why I think they are suited better for the detection process. Although part of the browser UI they are usually also not in the users' focus making them to some extent also prone to phishers adapting in case detectors are based on them. Although HTML-content offers a lot of different properties that can be used to compare different documents the huge problem of phishing page polymorphism will make it easy for attackers to adapt towards detectors being based on the markup that is invisible from the user. I hence recommend to use the visual channel for detection as it is in the focus of the users at all times [229]. For a detection method based on the visual content it will get much harder for the attacker to adapt to the detection method. Looking into the amount of noise that can be added to the visual channel of a website without the user noticing and whether that is enough to counter visual detectors will be subject to future research.

DR *What Detection Overhead and Thresholds are Reasonable?*

Answering this research question is very easy and yet impossible. The goal of any detector should be to have a false positive and false negative rate of zero as well as a runtime close to zero. However, this is not possible. The detection rate of today's blacklist-based browser approaches seems to be at about 15% of false negatives [219] or 85% true positive detection. As phishing is still on the rise this value does not seem to be high enough. In most cases the prototypes built in this thesis were also not able to outperform this value by far. The visual comparison prototype in project 5.8 achieved 92% of true positives without taking a specific threshold into account. Examples from the related work section reported even higher values. Common sense dictates that as long as no false positives are created by a detector each true positive result that can be found should be used to protect the user.

This might perhaps also be used as a recommendation for setting detectors thresholds. A threshold value should be chosen in a way that it results in nearly no false positives while still capturing as many phishing attacks as possible.

No matter which detector is used, filtering and dynamic whitelists can help to accomplish such a goal more quickly. In case of our project in subchapter 5.5 we used data type based filtering that already reduced the number of false positives to 22.4% of the original 100% input. Combining this with a dynamically growing white-list the false positive rate went down to 4.5% of all websites within one week – still without using any detector at all. Another interesting result was that the number of new websites a user visits having critical data involved, drops much faster than the number of new websites a user visits in general.

6.1.2 User Intervention

During my research I applied the same five research levels also to the process of developing and testing user intervention mechanisms. The following sections will summarize my findings for the five main research questions concerning this area.

ID *What is User Intervention?*

In the introduction section 1.3 I defined user intervention as the step after detection taking the results towards the user. Within the project of this thesis this mostly meant developing a suitable user interface to display the detector outcomes to the user and enable here to decide whether to stay on a website or leave it. In project 5.2, 5.3, 5.5, 5.6 and 5.8 we built user interfaces for different kinds of detector data and in measured their success.

Although the design of the user interface – or a warning – is a central aspect of user intervention it is more than that. The data coming from a detector might need to be preprocessed or converted before getting a sensible output for the user (e.g. the location data in project 5.2 was converted to a more user friendly map representation which helps the user a lot. Determining the right thresholds and feedback type from the detector result might also be subject to the user intervention method depending on the type of detector use (e.g. in the privacy enabled concept in project 5.8) whereas in other cases this judgment might already be done by the detector itself (see the second detector architecture in project 5.8). This example shows the tight coupling of user intervention systems and the detectors that in some projects of this thesis also led to a detector development based on a user intervention concept and not vice versa (e.g. projects 5.3, 5.5 and 5.8).

IH *How can HCI be Used to Enhance User Intervention Mechanisms?*

As a lot of HCI related work has proved (see section 3.3) security is not the primary goal which is why a lot of security advice is overlooked. In project 5.3 we experienced this ourselves. For most of our user intervention methods we achieved well working results after we had used a user centered design process using interviews (e.g. project 5.2), focus groups (e.g. project 5.7), paper prototyping (e.g. project 5.2), iterative development with in-between evaluation (e.g. project 5.5) and field evaluations (e.g. project 5.6). So taking user properties into account is very important to ensure that a warning is being seen, understood and finally that a correct action is taken.

When creating a user intervention mechanism I recommend such a design process. Starting off with user research or a literature review about usable security can already yield interesting information about best practices and real life situations. Focus groups within this thesis to gain further insights and feedback for general ideas or to generate new user intervention ideas with the participants. Expert interviews can help to find the corresponding security counterparts of what the user's understanding is. Is it possible to match both worlds within the planned user intervention method?

In case of the data type based approach (5.5) and the visual similarity based project (5.8) I started with developing a user intervention method sensible for the user before even considering the detector development. Especially within the project on visual similarity we achieved a high amount of correct decisions of the users. Using such an approach it is possible to develop a detector that on the one hand has the advantage to fulfill the needs of providing data that can be understood by the users. On the other hand it also gets hard for attackers to subvert it as they would have to change a central part of the spoofing. I argue that the visual content channel is one of the most important for the user and as the results of project 5.8 showed users can make extremely mature security decisions in case it is used to explain security issues.

IM *How can User Intervention be Measured?*

Evaluating user intervention methods has been performed in a lot of different ways within this thesis. In case of our data type based alert dialogs (project 5.5) we used several lab studies to evaluate whether people would respond "correctly" to our appearing warning dialogs. This means that we measured whether they would accept warnings pointing out dangerous websites and leave the site (true positives) and turn down warnings that appeared in error (true negatives). Whilst the first measurement makes it possible to see whether the user intervention method is able to protect users' in case of a critical website the second measurement makes sure that people are not overly scared by the user intervention method and still able to find detection errors. Comparing both measurements for a condition using our plugin and a baseline using only a standard browser made it easily possible to compare both types of user interventions against each other. For example the fact that participants found 55% of all phishing attacks within the first lab study of the project 5.5 compared to 14% in the control group proves that our concept significantly increased the user protection. We also measured these decisions within an online study in project 5.8 to compare two different designs of our warning. A big advantage of such studies is that using a lab setting the actual detection process can be decoupled from the measurement of the user intervention results by setting the different detection results as independent variables.

In case of the projects 5.3 and 5.6 I measured whether the user intervention methods were suitable to change the participants' security opinion towards the websites to a larger degree than the values that were measured for a standard browser configuration. If a user intervention method manages to influence the users' security opinion towards the right direction, a necessary step towards a more correct security decision is done. However, it is still necessary to prove that such a change in security opinion will make the user refrain from using dangerous websites and how strong the opinion change needs to be.

In project 5.7 we measured the participants' recall of security indicators like the URL to see whether they had noticed the indicators on critical websites. As related work showed that these indicators are overlooked by the users, a change towards more indicator attention would also be a necessary step to better user intervention. As with the previous measurement the exact connection between both cases still needs to be found in the future.

In a few cases we also used field studies for the evaluation of our user intervention methods (e.g. project 5.5, 5.6). One downside of this kind of measurement is that it usually needs to be done with a detector in place. This makes the measurement of the user intervention method dependent of the detector results and an overall decision measurement would only be a compound value. A second problem with such a kind of study is that phishing attacks only happen very rarely when looking at a small user sample and even if attacks happen it would be hard to extrapolate them from the recorded data without affecting the participants' privacy to a large extent. Instead I used field studies to measure other parameters a lab study cannot yield, for example: how many warnings appear in the real world? how do users interact with the user intervention methods after several days of use? (see project 5.5). The special case of a software rollout as a field study has been then used in project 5.6.

Although they are to some degree artificial I recommend using lab studies as a main evaluation methodology combining them with other measurements were possible. To still get valid results a lot of parameters within the lab studies need to be carefully controlled (see section 7.2 for a list of recommendations for evaluations).

In project 5.7 we also used eye tracking to find subtle unconscious user behavior towards the security indicators or user intervention methods on the website. Those measurements should always be correlated with self reported answers to see whether the participants realized where they have been looking at. Mouse cursor tracking is a cheap alternative to using an eye tracker.

IE *How to Enhance User Intervention Quality?*

When creating warning indicators for user intervention methods two different approaches existed up to now: non-blocking indicators and blocking dialogs:

So far related work has shown that non-blocking indicators most often fail to be noticed by the users, an effect that we also saw in our project 5.3. But non-blocking dialogs are not completely unnoticed. In project 5.6 we used browser Personas to communicate the security status in the whole browser background area and were able to influence the users' security opinion successfully. In the project using a combined status bar (see subchapter 5.2) we used different security indicators that took up a lot of space. Some participants saw screen real estate being wasted here and they did not want more than 5% of their screens to be occupied. In project 5.8 we used fully blocking warnings for cases when visual spoofing of websites has been detected. In case of our project 5.5 our warnings were triggered by and during user input. A blocking warning in such a case would have been immensely disturbing. For this reason I invented a third type of warning I called semi-blocking warning that opens a warning that blocks everything but not the current interaction. Blocking upcoming input but not interrupting the ongoing interaction combines positive aspects of blocking and non-blocking warnings. This semi-blocking way of user intervention performed significantly better than the standard browser. Together with the semi-blocking concept we also introduced other new properties that can be useful for warnings. The warnings appeared in-context of the users' actions (time- and location-wise) and tried to preserve most of the the current UI appearance were possible. Future studies will be necessary to find out which type of enhancement contributed in which way to the good result of the semi-blocking warning type.

Although all different types of warnings seem to work if applied in the right way I recommend to use blocking warnings for critical situations changing to a less disturbing form of warnings whenever possible. Semi-blocking warnings whenever its needed to block the process in the end but a immediate interruption is not suitable and finally non-blocking indicators for less critical situations and positive reinforcement. Creating non-blocking warnings that are noticed within the accepted boundaries of used screen space is challenging. I recommend using background imagery of other UI components as a solution for that.

Besides the type of warning (non-blocking, blocking, semi-blocking) a lot of other parameters play an important role. Changing imagery in user intervention methods seems to be well accepted by users as the map module in project 5.2 was highly accepted. In project 5.8 we even used screenshots of the similar websites as a main element of the user intervention and we found that even after repeated exposure to eight warning messages the interaction times with the warnings were not reduced. Within our focus groups, for example in projects 5.5 and 5.7 we gathered interesting design input on warning design. The warnings should be to some extent unique and form an own group of warnings such that the the underlying user intervention mechanism can immediately be recognized. On the other hand all warnings should look differently such that habituation is prevented and each new dialog is noticed again. This largely depends on the contained information in form of text, imagery and layout. Whereas the focus group results were in most cases congruent the focus group participants had different opinions on the color scheme of a user intervention. Some wanted very subtle colors in the style of the operating system and others wanted to have flashy colors to make a warning stand out.

In project 5.5 we used a green coloring of form fields whenever the user had previously entered critical data on the same web site to reinforce them positively. True negatives can also help to convey a positive attitude towards security as users don't experience that every time something security-related is displayed it might denote a catastrophe.

A last option for user intervention are community-based approaches (see subchapter 5.3). In the real world security advice is often acquired from other people but I experienced some hurdles when testing this concept on the computer: understanding security ratings is hard for the average user as they do not exactly know what a security rating is about. What does it contain or state? Another important issue was that we used solely aggregated security opinions of a community. In the real world inter-personal recommendations and security advice is given. These relationships would need to be resembled more closely on the computer (e.g. by including personal comments).

No matter which kind of user intervention mechanism is developed it always has to be combined with proper reasoning. The upcoming research question IR deals with this.

In project 5.7 we introduced errors into the content area of the web browser to make the users look more after the security indicators around. We observed that participants only got more confused about the content than it reminded them of looking towards other security indicators. This means that visual content of the website drags a lot of attention away from other more important security indicators, but reducing this effect is hardly possible.

In subchapter 5.5 we displayed warnings only on websites asking for critical data. This already reduces the number of websites that possibly need a warning to at least less than 23% of all websites and besides this provides a good reason for why the user intervention method appeared. Context-based filtering hence can also lead to more understandable warnings.

IR *When Should Intervention be Performed to Which Extent?*

Within this thesis I worked with all different kinds of warnings – non-blocking, blocking, semi-blocking – within many different user intervention methods. For each type of warning we achieved results that showed some kind of change towards more secure user behavior. But looking at related work habituation to warnings is an important problem that needs to be addressed by reasoning about the amount of warning users should be exposed to in different situations.

Especially in some of our lab studies (e.g. in project 5.5) where we purposely had a lot of consecutive warnings showing up, our participants complained about the high number of appearing warnings. As a contrast our web evaluation testing visual similarity based warnings had no such complaints and also had a constantly high viewing time of about 20 seconds for eight similar warnings in a row. This shows that it is hard to report exact measurements on when warnings start to annoy users and how this may lead to warning habituation in the future. To get more insights here it will be important to have a look at different parameters of warnings that could be able to slow down the habituation process.

Another possible point of reasoning is the level of detail that is used for explaining a security problem within a warning message. In project 5.5 we experienced that making a lot of different sub-detection results publicly available can also lead to confusion. If one sub-detector classifies a website as malicious while another one says a website is not and both is displayed towards the user how should a final decision be made?

From the research done within this thesis so far a possible recommendation could be that critical situations need critical warnings, but the more interrupting a user intervention method is, the worse is the habituation effect caused by false positives of the warning. In the worst case this error does not only lead to users distrusting this specific kind of user intervention but it can also lead to a general distrust in computer warnings per se. In case of less critical situations or if the underlying detection is not very accurate, a non-blocking indicator might be better suited than a blocking warning causing habituation. In some other cases interrupting the user's current task might not be a good idea (e.g. while typing). In those cases semi-blocking warnings can help by only blocking the completion of a final critical action but not immediately interrupting the user's current action. In case of the data type based user intervention other properties like the positioning of the warning also made it possible that the warning was less invasive and hence was seen as less disturbing.

In many cases the user intervention method has to define thresholds for the detector values that will finally fire certain kinds of warnings. These thresholds should always be tweaked towards having a really small number of false positives (see research question DR for details).

A researcher or developer of user intervention mechanisms should always think about what bad consequences could arise in case of overuse of a given intervention. Security protection is important but productivity and usability of the primary task has to persist. When designing user intervention methods it is important not to loose focus on this topic in favor of the formal success of the own method. User intervention always needs to be regarded in a larger context. In some cases researchers already proposed to get rid of all phishing protection to gain better productivity for the rest of the users not being affected by attacks [123]. There certainly is a valid point in this kind of related work but it mainly addresses the point that it is important to match the right balance of security measurements in favor of usability.

As mentioned before positive reinforcement of non-critical cases can also help to create a positive mood towards security and to make users being less frustrated about appearing warnings. Positive reinforcement should always be done using non-blocking indicators compared to blocking or semi-blocking approaches for the error cases. In the ideal case a missing positive reinforcement could already cause the user to perform security inspection by herself.

6.2 From Phishing To General Security

Although the projects and research findings within this thesis all have their focus on phishing detection and the user intervention thereof, it is possible to apply a lot of the findings made to general computer security and perhaps even beyond.

Other security detectors and disciplines should also look out for how a user centered design of user intervention methods and detectors can enhance the final results. Starting from looking at what a user can understand and then deriving a detector mechanism out of this.

Another valuable principle that can be applied on a broader scope is to look out for what actions users undertake when security critical things happen. Where is their focus? Can this focus be used to explain the security issue better? An example could be setting up Firewall restrictions the moment a new software component is installed instead of not being noticed at all or being noticed in case the software tries to access the Internet for the first time (perhaps without even having the user know). Mobile operating systems like Android[1] already ask for specific permissions a user has to grant its apps whenever they are installed on the phone.

The concept of semi-blocking dialogs could also make sense for other tasks. An office software for example could create semi-blocking notifications when asking to create a backup copy, or collaborative working systems could withhold changes other users have made up to a point when the user stops typing and focuses on the screen again.

These are only a few examples of the more generalized application areas of the research findings of this thesis. In many cases I already tried to phrase the findings in the previous subchapter 6.1 more universally.

6.3 Detector and User Intervention Model

Based on the aforementioned research findings I created a summarizing model that shows the different influencing factors of the detection and user intervention flow. Using this model it is easier to recognize important properties that need to be taken into account when building new detectors and user intervention methods for phishing. The model is depicted in figure 6.3.

The most important aspect is to understand detection and user intervention as an overall combined process where neither the detector nor the user intervention side can be examined without considering the respective counterpart. The global context and the user and attacker properties are influencing factors towards each system. Despite that the importance of the attacker decreases towards the user intervention methods (whereas the user importance increases) both parties have to be considered throughout the whole process.

The detection process can make use of pre- and post-filtering to reduce the result load that is passed on to the user intervention method. The detector itself can make use of a variety

[1] `http://www.android.com/`

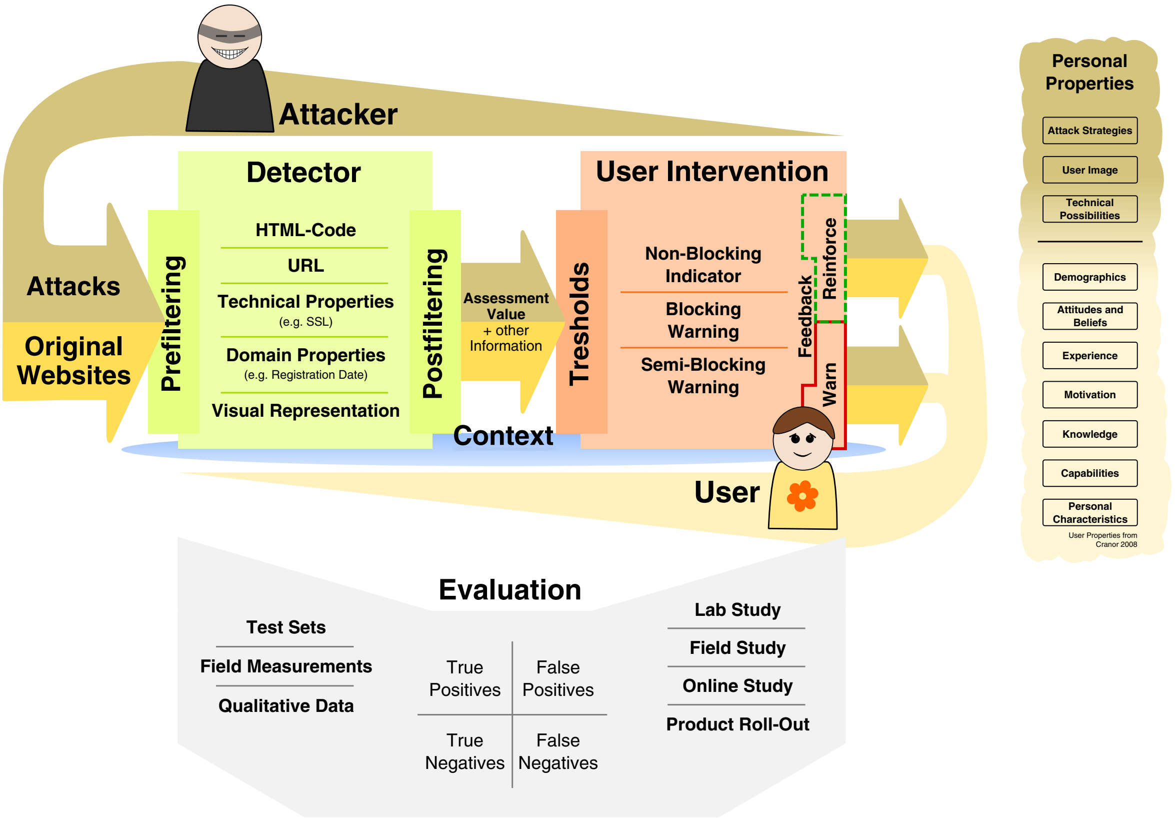


Figure 6.3: Model of the interplay of detection and user intervention, its important parameters, evaluation processes and stakeholders.

of inputs to calculate a final assessment value that is at best combined with a recognized brand. The user intervention makes use of the passed values and by using certain thresholds, warnings of different kinds can be displayed.

Both parts of the system need to be separately evaluated. For detectors test sets are most often used while a combination of lab and field studies makes most sense for the evaluation of the user intervention strategies.

Take Home Messages

➥ **6.1 Answers to the Research Questions:** Phishing detection and user intervention are closely coupled. It is not only possible to identify research domains that overlap but even central concepts are shared between the two areas. Definitions of detection and intervention can be made along true and false positive measurements. HCI and a user centered development process play an important role for both dimensions and a development process should take user intervention into account from the beginning or even reverse the development process to start with the intervention before the detector. Detectors that make use of user-distant properties might also work well but will most often allow attackers to adapt to the detectors. User centered detectors should hence be preferred.

➥ **6.2 From Phishing To General Security:** The general idea of combining detector and user intervention development with a focus on the user's attack models and the attacker's adaption possibilities will be easily applicable to a much larger field than phishing protection itself. Besides that a lot of other findings of this thesis are applicable beyond phishing.

➥ **6.3 Detector and User Intervention Model:** The interplay between attackers, users, detectors and user intervention easily fits into a straightforward model presented in figure 6.3. This model should be taken as a basis for the everyday development of protection methods.

Chapter 7

Recommendations and Guidelines

Getting anti-phishing right has been a goal for researchers and companies for many years now and the perfect anti-phishing method is still to be found. This thesis also does not contain a perfect final product but offers parts of the secret recipe for such a perfect system. I was able to find a large number of interesting findings and results that were discussed in chapter 6. During those studies I gathered a lot of knowledge concerning good and bad practices and I want to include them within this chapter.

Beginning with a utopia of how phishing detection and reporting should perhaps look like, I take another look on whether or not it seems at all possible to create an Internet without phishing and how the cornerstones of such a web would need to look like. I conclude this chapter with guidelines for conducting a good phishing user study by discussing lab vs. field studies, providing an optimum scenario for a lab study and giving an overview on statistical methods for evaluation of such experiments.

7.1 A Utopia of Anti-Phishing

Looking at the model of phishing detection and user intervention given at the end of the last chapter (see section 6.3) combined with the rest of the findings of this thesis one can try and guess how a perfect detector and a perfect user intervention method could eventually look like.

7.1.1 Achieving the Best Detection

The utmost perfect phishing detector would be a detector with a zero false positive and a zero false negative rate – every website would be classified correctly. Creating such a detector is rather impossible because websites do exist where the classification in malicious and non-malicious cannot finally be determined. Imagine a shady retailer offering faked clothing and storing the users' payment information on servers that are virus infected and the data itself can be downloaded by others. Would such a website by malicious? Would it be phishing? Perhaps even knowing what the website is all about, it could be that some people explicitly want to interact with such a website. This clearly shows, no matter how perfect a detector would be, some kind of final user decision will always need to be possible.

However, promising approaches for detectors have already been found. In related work many different technical parameters have been taken into account and have been analyzed by machine learning to come up with detectors with relatively high detection rates. Combining such an approach with other findings made within this thesis (data type based filtering, URL spell checking, visual comparison), it seems possible to get a detector that is able to take the right decisions for nearly all websites. To find the perfect balance between those features machine learning will be valuable at some point (see [1] for a machine learning comparison for phishing). A danger that has to be considered when using machine learning is the way attackers could adapt towards the algorithms – even machine learning needs to be secure [22].

Besides the high accuracy I found other properties that need to be taken care of when building such a detector. Is there an interface of the detector results that can be used for the later user intervention mechanism? Will the users be able to understand and heed the warnings, and will they be able to still detect false detection results in case some exist?

A second important aspect that needs to be considered is what would happen if attackers change certain parameters of their attacks. In case the adapted attacks would get through the detector without loosing their effect on the user the detector has to be changed to be able to handle such evasion attempts. Assuming a perfect detector for phishing attacks to exist and to be available to web users, the only way for attackers to continue would be to add a large amount of random noise to channels, which in turn would be relatively easy to be recognized by users. In other words this means, if high quality phishing attacks are detected by the detector the users will hopefully recognize or ignore the low quality attacks that remain.

In summary this is why reversing the development process and starting off with the user intervention methods will often lead to detectors that automatically lead to working detectors that last. In case a good detection accuracy can be achieved using such a user intervention method, the result will be close to perfect detection.

Throughout this thesis I have used many HCI related observations to come up with my detector ideas. For future detectors valuable methods for finding new detection possibilites will be the following: Looking at how security problems are dealt with in the real world, how attack strategies of the phishers work or how users judge trustworthiness and understand security.

Finally let me mention the potential introduced by pre- and post-filtering of detector input data. In case a detector needs high computational load, filtering can be used to get rid of cases that need no computational check for different reasons.

7.1.2 Optimal User Intervention

The three main types of user intervention (non-blocking indicators, blocking warnings and semi-blocking intervention) have been extensively discussed within this thesis and they all add their share to good user intervention in different contexts.

An optimal user intervention method would make use of those different concepts in a way that the indicators are correctly noticed without annoying users by their existence. Non-blocking indicators could communicate positive security reinforcement while blocking warnings are used for the definite problems found by the optimal detector described above. If applicable, semi-blocking intervention could be used as an alternative.

The general design properties of an Internet browser warning have been changing throughout computer history from small gray dialog boxes popping up in the center of the screen, to small yellow warning bars that unfold in the web browser on to red colored warnings similar to websites in the content area. These designs couldn't be more diverse and we could confirm this during the focus groups for our different projects. Some people see decent gray colors as being more appropriate for an operating system and that flashy colors would just look like advertisements. Others think that a warning needs to be flashy to stand out.

My personal thoughts here are that in general there is no optimal solution to this problem as warning messages need to be something special the user is not habituated to. Hence, changing the design from time to time can help in keeping up the interest in such dialogs. The more the habituation causes (e.g. false positives) are avoided the longer a warning design will last.

Habituation effects can be further reduced by using warning contents that contain many unique and understandable elements within the warning dialog. Within our project concerning visual similarity (see subchapter 5.8) we made use of large screenshots and held the user concentration towards our dialog up during a repeated exposure to eight similar looking warnings in a row.

A perfect user intervention will bother the user only if necessary and will in those cases supply her with the best information necessary to quickly understand the situation and take the right decision. Such understandable pieces of information can be for example screenshots of involved websites or maps showing a security problem. In case of security critical original websites a positive reinforcement of the user can be used to not only amplify the trust in the current action but also mitigate trust in case the reinforcement is missing in future situations.

7.1.3 Future Proof Methods

Security is usually about playing cops and robbers. Whenever security experts bring up a new technology to protect the users from a threat, attackers are starting to develop counter measures and vice versa. So how can we make sure that the steps illuminated in this thesis will not only work until the next type of attack is developed?

In fact it is not possible to guarantee this as new technologies might always arise that allow new forms of phishing. However, taken the rules above into account the developed anti-phishing mechanisms should be quite reliable. The way users perceive and judge things and the possibilities of social engineering haven't changed a lot since their very beginning. Only the used technology and the applied principles of the attackers have changed. Using the proposed findings for future protection, they should be mostly free from technological singularities and should focus on the cornerstones where social engineering targets the users exactly.

At the moment it is extremely easy for attackers to impersonate companies at points where users perceive and judge the trustworthiness of a company. In case this easy impersonation is hardened the number of phishing attempts will durably decline no matter which technology is behind.

If everything else fails the way of developing anti-phishing measures as explained within this thesis should still be applicable to new kinds of technologies and threats.

7.1.4 A Web Without Phishing?

While the previous subchapters took a future look towards detection and user intervention which was in the center of this thesis I also want to step back a little to look again at the future of the bigger picture of phishing. All anti-phishing methods that have been presented in the last decades have one issue in common. None solves the issue of phishing protection completely and none managed to significantly bring down the number of phishing attacks that exist. So will it ever become possible to create an Internet without phishing or should one just cope with the status quo and make the best of it?

From a more economic perspective to stop phishing completely it would be necessary to lower the profit that can be gained from phishing below the costs phishing causes (Schechter and Smith [258] looked more detailed into this). These costs are in general relatively low: hosting can be done using free webhosters or hijacked servers and only the personal spare time of the attacker is needed. The biggest efforts in phishing are to reach potential victims and finally monetize the collected confidential information.

In fact several ways could lead to a phishing free web by achieving to raise the costs over the revenue: When imagining a well working anti-phishing detector and user intervention system that would drop the number of gained correct credentials close to zero, the costs of phishing would be no matter what they actually are higher than an incoming revenue of about nothing.

Changing this cost vs. revenue relation can perhaps also be achieved by other means affecting different steps of the whole phishing process (please refer to figure 2.2). Reducing the possibilities for email spoofing for example, would also effect the phishing process as it would get hard to sent out impersonated email messages.

Besides this, changes in the technical properties like HTTP, DNS and HTML could also help to make it harder to impersonate other websites. Using a different access structure to the web instead of URLs could for example help to make it easier for users to recognize the case when they are visiting an impersonating web service instead of the original one – the Convergence Research Project [107] for example proposes such an architecture. Another possibility is proposed by Markham [177] using image hashes together with a domain name for easier recognition of typing mistakes. A few researchers work on methods to counter identity theft with digital uniqueness [223] or better password technologies [326] to achieve such a goal.

In many of these areas research is carried out to find solutions for such problems. However, the scope of this thesis was limited to the detection and reporting of browser based phishing.

7.2 Evaluation Recommendations

As already stated in the related work chapter (compare section 3.7) user study methodology is a huge area within usable security research. Planning, executing and evaluating usable security tools is especially hard due to different reasons. One example would be that security may not be the given primary goal of a study although it is in the center of evaluation. In table 3.3 I already gave an overview over different aspects of different usable security studies. Within this section I want extend on that and provide some advice about best practices I have developed throughout the different studies concerned with this thesis by combining them with methods that have been approved in related work. Within this section I will refrain from including backward references to the projects that yielded these insights.

Most of this section refers to user intervention user studies as the testing of detectors is much easier and straight forward having the right test set (see subchapter 5.1).

7.2.1 Preparation

When trying to explore user behavior on phishing or when evaluating anti-phishing tools for the web browser, many different aspects have to be taken into account already from the beginning of the preparation of a user study.

Real World Vs. Lab Studies

One of the most important basic decisions that have to be made is whether to test a concept in the lab or in the field. Field studies to evaluate anti-phishing concepts are extremely hard to conduct (including ethical and legal reasons). However, in addition to a lab study, it is a good idea to conduct some field testing that does not rely on actively induced attacks [183]. Even if one could create a study with a large number of participants, the number of occurring attacks would still be too small for significant assessments. Additionally, finding those attacks would require an enormous amount of data logging and post-processing – only Florênico and Herley were able to measure such values once [92]. Field studies may still be useful when trying to measure other variables besides anti-phishing success.

Lab studies in contrast allow for a precise control over when and how phishing attacks will happen and can also guarantee that the setting of the whole studies remains unchanged. A problem here is that the study environment may bias participants [271]. Hence, many precautions have to be undertaken to avoid such effects wherever possible.

Comparison vs. Direct Collection of Data

A lot of the studies conducted in the field so far have been dedicated to evaluate how and why people fall for phishing. In some of those cases, this restricts research to collecting direct data only – e.g. what percentage of phishing websites is detected. For other cases it is possible to collect data for two different cases and compare them against each other – e.g. comparing one anti-phishing concept against another one. Since the lab situation in itself may have a certain influence on peoples' behavior, the success of a concept should not be measured as a single value but as a comparison against a control group where possible. A lot of studies were conducted in this manner [78, 126, 324, 325]. In case the lab setting has had any influence on the study results it then would hopefully have influenced the control group to the same degree. This means that in a comparative study, the result of the comparison should still be valid even if the absolute number of detected phishing websites for a study is potentially higher than in a real world setting.

When comparing new concepts, three different options exist: Comparing two different versions of a concept, comparing against a similar concept that has been developed in the past as part of other research or, if no similar tools for comparison are available, to compare against a standard browser as a baseline.

Between vs. Within Subject Studies

Research that does not compare two concepts directly often uses a within-subject approach for the different independent variables [67, 126, 166, 294, 324]. For work that compares concepts against each other a between-subjects or even a mixed-design – combining both – has to be used [64, 157, 183, 280, 325].

Using a between-subjects approach, the baseline and the new concept are distributed among different participants. That is, the same websites, data, etcetera can be used for both conditions without causing learning effects. Another important point is that when switching between two different concepts for one participant – as it would be with a within-subjects design – the concept would would get much more accentuated than having it silently run during a whole session. Having a second control group using a different concept makes it also possible to gather their opinion on the concept without them actively using it. This can result in interesting findings compared to the participants that actively used the concept.

Websites

Please refer to the extensive discussion of possible test set websites in subchapter 5.1.

The Scenario

Security is never the user's primary goal [296]. Therefore, it is important to hide the real purpose of the study from the users before they complete all tasks. Whenever telling participants beforehand that the study is about security, the results will change dramatically [166] and all results can just be taken as an upper bound of what the participants would be able to detect in the best case scenario. Computer or website usability are topics that are often used to disguise the real purpose of the study in related work [72, 294]. Other bogus topics used can be found in table 3.3 in row 3a. The real purpose of the study should always be revealed afterwards [89].

During the study, participants will be usually asked to expose data at some point to validate whether they would have fallen for an attack or not. This data can either be their own real world data or some imaginary information provided to them in advance. This role-playing technique is often used but has been criticized for changing the result of the study [259]. Apart from pure authentic information and pure role-playing, there is a third option we call "diverted role-playing".

Authentic Data: Using the user's real data during the study is the only way to guarantee that the participants will be worried about losing this data to an attacking party. But this technique also has several disadvantages: It is always possible that the participant does not have the type of data needed (for instance, someone not owning a credit card). This would then disqualify her in participating in the study. Apart from this, anxious people might drop out of the study due to the fact that they have to provide their real information. Those people are an important user group for security studies. In the worst case, the remaining people will have the tendency to trust the lab situation more, reverting the effect that should be achieved using authentic data. Whatever way is used, the participants' demographics and the perceived risks should be as real as possible [259].

Role-playing: In a role-play study, the participant is told to impersonate the identity of a third person that usually does not exist. She is handed out information about the role she is going to play, together with information about this person. Role-playing studies are often criticized for the fact that the participants do not care for the data as much as they would for themselves because they know that they are just playing a certain character [259]. Another problem here is that since participants are told to change their own perspective, they will probably stop using their own set of ethical values for completing the tasks.

Diverted Role-playing: A way to solve this problem is the "diverted role-playing" technique as used in several studies [183, 324]. Here, the participants complete the task for a third imaginary person – usually some close relative, like the grandmother. They are just given information about that third role but use their own decisions and set of ethical values to solve the task in the interest of that character. In our "grandma is ill"-scenario we told the participants about their grandmother being in hospital for a couple of days and that they should take care of some important online transactions for her.

Participants

When selecting participants, it is hard to draw a perfect sample of the target group for phishing attacks. Usually, the number of security experts should be as small as possible and any kind of security study should not be conducted with participants recruited at the security lab. On the other hand, using arbitrary participants from the street is a potential problem as well. Their degree of Internet security knowledge will usually be very low but perhaps they do not use email or the Internet and are therefore no tentative phishing victims. A sample of participants should be most similar the average Internet user that is exposed to phishing threats from time to time. Not all papers published in the past reported on this or took special care but some explicitly state their recruitment process for the participants [78, 139].

The number of participants is always dependent on several factors like the number of independent variables. A dozen people for a condition may already be enough to get significant results and will usually also make it possible to assess qualitative feedback besides the quantitative measurement. With some studies being conducted online, a larger sample size can be achieved. Conducting studies in such a way still makes it very hard to ensure identical conditions with every participant besides the selected independent variables.

Whatever the degree of the security knowledge of the participants is, they should be evenly distributed to the different subject groups based on that criteria. Since the real purpose of the study should not be told in advance, it is important to assess this with other questions. Asking for "internet knowledge" instead of "online security knowledge" for example might be a way although the knowledge level can still differ.

Collecting demographic data in advance may help to distribute or refuse participants before the actual study is conducted. After the study – when participants have been debriefed – it is then possible to ask people directly for specific demographic data on security and phishing or measure this explicitly. The collected data can be used afterwards to see whether the distribution was correct or which other demographic factors might have had an influence.

Tasks

Since it is important that participants do not know the real purpose of the study in advance, the experiment needs tasks that fit the fake purpose and the scenario that was selected. For a study that is supposed to be about “website usability” this could be several shopping tasks. Tasks should also be built up to quickly lead to the actual point of testing without making the fake purpose too obvious. For a shopping task, not the selection of the product is the critical part but the entry of personal data. Existing phishing websites can provide insights on how to do this.

Dummy tasks: To be able to disguise the study purpose, it is important to include several dummy tasks that make the scenario more believable. Usually, those tasks should be similar to the attacking tasks to keep them from standing out. To be able to create a proper balancing (see upcoming section) the number of dummy tasks need to be multiples of the the number of phishing tasks – this means that usually at least half of the tasks will be dummy tasks. To make sure that those tasks do not have any effects on the study outcome, they should be balanced together with the attacking tasks. To achieve this, a possible solution would be to create two fraudulent tasks per level that will be measured. Secondly, for each of the fraudulent tasks, a legitimate version is needed. Having this portfolio of websites, it is now possible to present one legitimate dummy task and one attacking task per level whilst it is still possible to balance everything. Please refer to the section on balancing below for more details on how to achieve this.

Real Interaction vs. Images: A lot of studies just use screenshots instead of a real web browser setup that the participant can interact with [72, 126, 261]. This increases the artificiality of the lab setting since people are not able to interact the way they do at home. A participant that watchfully checks for phishing websites could try to do many things to verify the genuineness of a website, like checking the SSL certificate details, or setting up an extra search for that company. If the target of the study is beyond just assessing the visual impact of a screenshot, a real environment has to be preferred.

How to Get People to the Websites

Discovering phishing attacks can happen in many stages of an interaction with data from the Internet. Using emails with links, for example, may already produce dropouts because people are going to notice the fake URL already at this point. In case a browser-based concept is compared to another browser-based concept, the dropouts due to link detection in an email will not matter because they would be more or less equal. In such cases, it is possible to use bookmarks during the study that the user has to visit, or by simply remotely opening the websites for the user. This reduces the number of interception points that are independent of the concept that is tested.

7.2.2 Ethics and Privacy

In many countries, studies have to be approved by an Institutional Review Board (IRB) before being conducted. Other countries just have the rules for this kind of research laid out in the law. In any case, one has to take care of doing the best to anonymize the collected data of the participants and to protect them from potential harm through phishing websites. Whatever steps are taken to ensure proper privacy of the participants, should be documented and published together with the results of the study itself. Finn et al. [89] describe this in detail in their paper.

Whenever a study purpose is fake, it should be made clear to the participant after the study what the real purpose of the study was and all questions regarding the process should be answered. For the research, this again can be used to get valuable input from the participants on how the study looked like from their perspective.

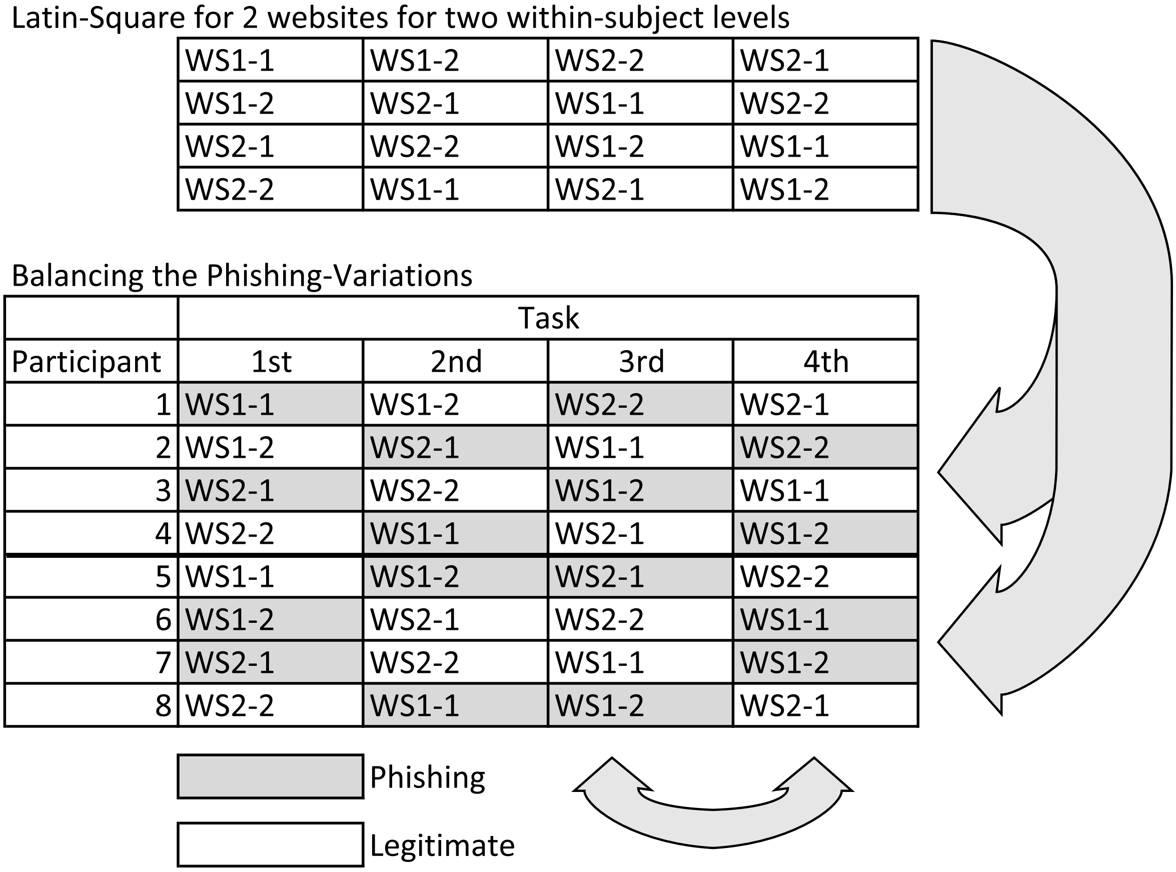
7.2.3 Execution

When the study is finally performed, a lot of new questions arise. How to technically conduct the study? Which hardware to use? How to balance the study results?

Balancing

As already stated in the preparation chapter, it is extremely important to have a balanced study setup. This makes sure learning effects in case they occur may not influence the study results. As mentioned earlier, for each website used in the study one phishing and one real website should exist. Since each participant needs to have at least one legitimate website and one attacking website for each part of the concept that is tested this results in four different websites that need to be prepared. An example: If a banking website should be part of the study and the chosen brands are for example “Barclays” and the “Bank of America” (BoA) one participant would get the pair of a legitimate Barclays site combined with a phishing version of BoA, whilst the next participant would see a phishing Barclays version with the legitimate BoA website. To make sure the order of the two websites did not induce any effect it is hence also necessary to reverse the order of the two websites. Using more websites makes reordering more complicated.

With this rule, one can already compute the minimum number of required participants per group. Basically this is $P_g = n_l * (w_p + w_l) * 2$ (with n_l being the number of within-subject levels to test; w_p the number of phishing websites per level and w_l the number of legitimate websites per level). Using one phishing and one legitimate website ($w_p + w_l = 2$) this results in the fact that the number of levels has to multiplied by 4. If multiple concepts are tested in a between-subjects study P_g should be used for each of those groups. The number of tasks to perform will always be half of this. To get a good task order for the minimum number of participants a latin square [109] can be used two times inverting the phishing attacks in the second set. Figure 7.1 shows an example for a possible task order for two phishing and two legitimate websites, used to test two different within-subject factors. This task order is then used for each of the concepts that are tested between-subjects.



Latin-Square for 2 websites for two within-subject levels

WS1-1	WS1-2	WS2-2	WS2-1
WS1-2	WS2-1	WS1-1	WS2-2
WS2-1	WS2-2	WS1-2	WS1-1
WS2-2	WS1-1	WS2-1	WS1-2

Balancing the Phishing-Variations

	Task			
Participant	1st	2nd	3rd	4th
1	▨ WS1-1	WS1-2	▨ WS2-2	WS2-1
2	WS1-2	▨ WS2-1	WS1-1	▨ WS2-2
3	▨ WS2-1	WS2-2	▨ WS1-2	WS1-1
4	WS2-2	▨ WS1-1	WS2-1	▨ WS1-2
5	WS1-1	▨ WS1-2	▨ WS2-1	WS2-2
6	▨ WS1-2	WS2-1	WS2-2	▨ WS1-1
7	▨ WS2-1	WS2-2	WS1-1	▨ WS1-2
8	WS2-2	▨ WS1-1	▨ WS1-2	WS2-1

▨ Phishing

□ Legitimate

Figure 7.1: Example for balancing two within-subjects levels using four different websites – all available as legitimate and phishing. A Latin-Square is used two times switching the last two columns in the second set and inverting phishing websites.

Measuring Detection Rates

During the study, the experimenter has to measure the number of fraudulent websites that have been detected by the participants. It is very important to lay out strict rules what counts as detection and what does not count as detecting an attack. Those rules should be reported together with the results of the study. In real life, a simple thought of “that URL looks weird” might also not be enough to stop the user from entering data on a phishing website. In fact, the experimenter must neither encourage nor discourage participants in any step of the decision process but it is important that the participants know somehow that they have the option to drop a task. This should be made clear during the introduction of the scenario without emphasizing it too much – otherwise it could again switch the user’s focus to security.

A good threshold for having detected a phishing website is when the participant has undoubtedly pronounced the website as attacking or aborted the task. In case the user makes any comments that she has any smaller doubts those can still be recorded for later qualitative reporting. There should be no obvious possibility in the user study to report fraud as this might bring security to the user's focus.

True and False Positives Besides measuring the number of correctly identified attacking websites (true positives), it is also very important to count the number of false positives – the number of legitimate websites that have been accidentally reported. If only true positives are measured or reported, a concept that is just frightening enough will receive high true positives and hence look good. In fact, it would also scare people off using legitimate websites which is an important fact to report.

Hosting the Study

Independent of whether the legitimate websites are coming from real servers or not some kind of fake hosting has usually to be set up to divert the browser during the study to fake websites. Please refer to subchapter 5.9 for an extensive discussion about hosting websites for a study.

Data Recording

Besides measuring the absolute performance of the concept and additional questionnaire evaluation, more data can optionally be recorded.

Video or Audio recording: Performing video or audio recording of the participants helps to identify smaller stages in the decision process more clearly after the study was conducted. The problem of video recording usually is that the user feels even more monitored than with the sole presence of the experimenter.

Eye tracking: More subtle and sometimes even more accurate recordings of the participant's behavior can be done by using eye tracking hardware. Modern eye trackers can be perfectly integrated into the study setup without disturbing the participant. Especially when it is important to measure whether a specific screen region and hence a specific feature of the concept was noticed by the participant or not, eye tracking may be used. Whalen and Inkpen used eye tracking extensively in their study [294]. Due to the expensive hardware, mouse tracking can sometimes be used as an alternative.

Mouse Movements: Mouse movements of a participant often closely relate to the eye movement of that person [46]. Especially in the browser, it is easy to set up mouse movement recording in the background [18]. The data gathered here can also give advice which screen regions were considered by the user for reading or clicking and which ones were not.

Hardware and Software Setup

Although web browsing in general does not have any high hardware requirements, the hardware setup of the study might be of importance. Especially screen size and screen resolution influence how large the security indicators will physically be on the screen which can correlate to how much they will be noticed. Input devices can also play an important role as this may influence how much time the user can spend reading screen content. In any case, the whole study has to be conducted using the same hardware setup and it should be documented to be able to report on it later.

It is possible that the participants are not used to the specific setup in terms of operating system or to the type of web browser. Instead of providing every participant with the setup she is used to, we recommend to simply use the most common configuration the average participant would have while asking every single one for their standard configuration. Having their usual configuration recorded makes it possible to rule out significant dependencies due to the unaccustomed setup afterwards.

7.2.4 Analysis

After having conducted the study, the data needs to be analyzed. Firstly, this makes it possible to report on the participants' performance for the different levels of the independent variable. With different statistical tests it is also possible to find additional effects. This section reports on the different analyses that can be made after the study has been conducted. Before applying a specific statistical test it should be checked whether it is really applicable towards the current data set. In many cases inferential statistics are not correctly used in HCI [40].

False and True Positives and Negatives, Accuracy and Precision

Besides the qualitative data that has been acquired before, during or after the study using questionnaires and semi-structured interviews, the main dependent variable that is quantitatively measured is the number of attacks that have been detected by the user or in other cases a detector (true positives). As mentioned earlier, it is also important to keep track of false positives that have occurred (cases where an original website has been misclassified as being phishing). Please refer again to section 1.4 in the beginning for details about these terms.

The ROC-curves diagram and my false positives/negatives plot are simple diagramming techniques to see detector results in dependency of different thresholds (see section 1.4).

Statistical Tests

Having those values, the results of the analysis can be checked for statistical significance. Depending on the type of the study, the study design and other factors, different statistical methods have to be used. The ones most commonly used in past experiments are shortly explained here. This chapter is not though of as a replacement for a textbook on statistical test but is rather meant to report a list of statistical tests that turned out to be especially useful for evaluations within the area of this thesis. The data collected throughout my experiments is usually parametric. Hence, parametric statistical tests are used for statistical analysis of the data. In fact many statistical tests do not work well for boolean outcomes that have been measured (e.g. detected a phish or not). For each test this should be checked in advance.

t-Test The t-test is used for comparing two experimental situations and has the ratio between means divided by an estimate of the standard error as result [87]. Depending on the type of design – between-subject or within-subject – the independent or dependent t-test is used. In the evaluation of "WebWallet" [325] a t-test has been used to compare the standard browser and WebWallet condition.

ANOVA In case an independent variable has more than two levels, the t-test does not work. Here, an analysis of variance (ANOVA) is usually used. In this case, the null hypothesis tests for the same mean through three or more means [87]. If this test has a significant result, it is still not sure which level of the independent variable had which importance. Post-hoc testing is needed for that. A two-way ANOVA can also be used if multiple independent variables overlap. An example here is the moodyboard study where two variables where tested using ANOVA [64].

Chi-Square Test For the analysis of categorical data, other ways have to be used. Pearsons chi-square test can be used for this [86]. Downs et al. [72] put the responses and the properties of their participants in categories and hence use the test for the analysis. A problem with this test is that it assumes data to be near the chi-square distribution and hence needs large samples (232 participants in case of [72]).

Fisher's Exact Test Fisher's exact test computes its own chi-square probability for the provided data and can therefore be used on smaller samples too [86]. Egelman et al. [78] had some results in their categorical analysis that appeared less than five times. They used this test for the categorical analysis.

Pearson's Correlational Coefficient To measure the relationship between two variables and to find out whether they are associated, the Pearson Correlational Coefficient can be used [86]. Dhamija et al. [67] used this to measure whether the age of participants correlated with the participants' scores.

Cochrans Q test Combined with McNemars Post Hoc Tests Cochrans Q test is more or less an ANOVA for binary outcomes with multiple levels. As it is an extension of the McNemars test it can be used for post-hoc testing of the separate level dependencies [86].

Take Home Messages

➥ **7.1 A Utopia of Anti-Phishing:** Relying on the findings and methods of this thesis and of related work found so far perfect detectors and a perfect user intervention seems achievable and future proof. In a joint development process detectors that are close to the user's understanding of security and her mental models can be built that automatically resist the efforts of attackers to adapt towards the detection. In the optimal case this will lead to an economic state where phishing is not profitable anymore and phishers will have to move on to other types of crime or hopefully become faithful.

- **7.1.4 A Web Without Phishing?:** If phishing becomes uneconomical for the attackers will they refrain from phishing in its current way but will that last forever? Although technology is moving fast and new technologies offer new possibilities for different tactics, phishing is in the end a social engineering crime that exploits the same models of human behavior since its very beginning. In case protection is based on these "social weaknesses" of the user the protection should be mostly free from the technical background.

- **7.2 Evaluation Recommendations:** Conducting user studies around usable security is a hard thing to do especially as security must not be an obvious aspect of the study. Throughout preparation, execution and analysis a lot of important steps have to be planned and executed. Before conducting such an experiment the different aspects should be checked against existing recommendations as the ones presented here.

III

Conclusions

Chapter 8

Conclusions and Future Work

As the last chapter of this thesis I want to take a short look back on what was covered, what questions were asked and finally answered and where the journey of phishing protection, detector and user intervention development and research in this area could perhaps move on to in the future. A lot of this has actually already been covered in the chapter 6 and chapter 7 and won't be repeated here. Instead, only a short reference to the most important findings and a list of open points that lead to future research topics is presented to conclude this thesis.

8.1 Summarizing This Thesis

This thesis was based on two main and major issues: on the one hand the immense problem of phishing attacks as a social engineering technique that produces loss of millions of US dollars each year and on the other hand the growing field of usable security that tries to bring together two research disciplines which were formerly disconnected but now are becoming more and more closely interrelated.

In chapter 2, I looked closely at the phishing problem to lay a basis for the upcoming chapters. What exactly is a phishing attack? What is its history and its current state and how do current browsers cope with these attacks? From all the different steps of a phishing attack I chose the phishing encounter of a website in the browser as the subject of my research as this is a bottleneck where most of the phishing attacks are finally carried out and where changes can be most easily made and tested. The related work chapter (see chapter 3) afterwards reported numbers and research about the phishing problem itself, about the reasons why current systems fail and why users are blinded by the attacks before I finally presented prior solutions in terms of detection and user intervention.

Taking the general attitude for this thesis that detection can be only effective together with user intervention and vice versa, a twofold set of 10 research questions in five different levels has been brought up (see chapter 4). Within nine different projects described in chapter 5, I used a multitude of different methods and techniques to find answers to subsets of the research questions within each project. Besides the research findings and prototypes that were developed, I also presented completely new approaches and ideas like pre- and postfiltering of results based on the user context, using semi-blocking dialogs to reduce frustration, or a new technique of "diverted role-playing" for better evaluation.

Before presenting those findings to guide future development and evaluation in chapter 7 the answers to the different research questions were summarized in chapter 6. This included basic but extensible definitions of phishing detection and user intervention; showing how HCI could be used to enhance both, detectors and user intervention mechanisms and their quality, providing lots of examples of parameters that can be measured throughout both areas and finally reasoning about how much of effort is worthwhile in these areas.

These findings could not only be combined into a final model (see section 6.3) but many of them can also be applied to other domains of security detection than phishing alone (see section 6.2).

Looking at the research results that have been proposed by other researchers and me within the last years many promising ideas have been presented and proven to be successful but as prototypes alone they do not have the power to counter the real world attacks out there. Companies at the edge of the process (e.g. browser vendors) need to get hold of these ideas and need to deploy them within their systems such that they can reach the customers. Offering them as a downloadable upgrade is sadly not enough, as security is not the users' primary goal and most of them hence do not actively look for security software.

8.2 Open and Future Work

In the end the philosophers stone of phishing detection or anti-phishing in general is yet to be found and although I presented interesting concepts within this thesis more concepts should be developed in the future, at best by taking the recommendations given in this thesis into account. As outlined with in this thesis building a perfect technical detection mechanism alone is close to impossible but developing a phishing protection combination that makes phishing unprofitable is in my opinion possible when using the right means. I explored the interplay of phishing detection and user intervention using various methods to a large extent which already created a huge amount of possible future research in that area.

In section 5.8 I introduced the equilibrium value of false positives and false negatives of a detector and used it to compare different detectors using one single value. In how far this measurement really is a good measure for detector quality and whether it can hold for other application areas of detection will have to be verified in the future. Looking at evaluation at a broader scope it would be interesting to take a deeper look at evaluation techniques in general and which kind of evaluation techniques fit best towards phishing detection. Another part of the evaluation is the bridge between detector and user intervention evaluation. As already stated in the results chapter if field studies are used user intervention evaluation measurements may be influenced by the detector performance itself. Future research should take a closer look at this interplay to propose even better possibilities for a continuous evaluation of detectors and user intervention methods.

In several sections – especially in project 5.4 – the quality of phishing websites played a role within the detector evaluation. The definition of “phishing attack quality” and the role of attack quality within the whole deception process would be a very interesting field for future research. How should phishing website quality be exactly defined and what impact does it have on the success of phishing attacks. Finally the potential arising from this measurement to built better detectors should be taken further than it was done within this thesis.

Within the user intervention methods in the different projects of this thesis we presented a wealth of different additional informations within the user interface dialog (ranging from technical properties – like SSL encryption status – to visual screenshots of the websites involved within the detection process). For future analyses it would be interesting to find out which kinds of additional information are best for a user intervention dialog in terms of habituation and correct decisions. What information is understood by the users and how do they make sense of different presentation styles of such information?

Using the visual representation of websites for detection was discussed in project 5.8 and found especially useful as this information could not easily be changed to avoid detection without the user noticing. Future studies should take an exact look at how this interplay works. So how much noise needs to be added to a visual representation of a website that a detector based on that information will be effectively fooled and how will a user then really notice the changes to the website or would she still fall for the attack? Such tests could back up the general idea of this thesis and clarify the exact dependency between both. As we already saw in the subchapter 5.8 the image comparison method chosen is important for the success rate of the whole approach. Comparing screenshots against each other is in many aspects different from classical image or photo comparison. Perhaps a special comparison method can be found that focuses on the properties and problems of such an approach.

In some of the projects (e.g. 5.6) I measured the change in the users security opinion instead of direct phishing website behavior. With the presentation of such an alternative measurement it should be assessed in future work how large a change of security opinion of a user needs to be to actually stop them from performing a dangerous action and whether this measurement really can be taken as an identifier for user intervention performance. More or less the same holds true for actions that try to bring more user attention towards existing security indicators (e.g. what we tried in project 5.7). How can the users’ attention towards existing indicators actually be raised and does such an additional arousal in the end lead to a more secure behavior? Perhaps this might also be dependent of several factors like the security indicator itself in combination with the awareness that it receives.

Within this thesis I introduced the new concept of semi-blocking warnings (c.f. 5.5) which was successful for a special application area but introduced several new parts of a warning dialog within one concept: semi-blocking, in-context appearance, preserve context. Future studies on that type of user intervention should find out how these different parts relate to each other and how each of them add to a well working warning dialog.

Another topic that plays a big role within user intervention and that could only be tackled to a small extent within this thesis is habituation towards warning dialogs. Although its existence has been clearly proved in related work and I could confirm this effect in many projects, it seems to be possible to prevent habituation with a good user intervention method (c.f. the user intervention evaluation in project 5.8). For future research it would hence be interesting to see what exactly causes habituation towards such dialogs and how habituation can be reduced. Furthermore, does habituation towards one kind of warning affect other kinds of warnings? This will certainly play together with the overall warning experience a user has on her computer. Which kinds of warnings are experienced as belonging together? What warnings are experienced as useful?

An important future improvement for research would be a more consistent way of evaluating detectors. As already explained in the results section 6.1 at research question DM, a testing framework that would standardize test set collection and use between different detectors would be great to achieve more comparable results with less effort.

Future research should also include the more neglected paths of development and evaluation, perhaps by focusing on the personal properties of an attacker or by conducting more evaluations using field trials [24] or software deployment. The most correct measurements for anti-phishing software would be to generate success-reports of the real field use – although this is very hard.

Concluding there is a lot to look at in future studies whether it is to look more closely at a certain property of phishing detection and phishing user intervention or at detection and user intervention within a broader scope and application domain.

8.3 A Final Take Home Message

Usable security as a tug o' war between security and usability research is a great field to learn an important research lesson that is applicable throughout all research domains: one should never stop looking beyond one's own nose to see where the own research is embedded in. What is a perfect security detector worth if no user takes the advice?

IV

Bibliography

BIBLIOGRAPHY

[1] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair. A comparison of machine learning techniques for phishing detection. In *eCrime Researchers Summit*, eCrime '07, pages 60–69. ACM, 2007.

[2] M. Aburrous, M. Hossain, K. Dahal, and F. Thabtah. Intelligent phishing detection system for e-banking using fuzzy data mining. *Expert Systems with Applications*, 37 (12), pages 7913–7921. Elsevier, 2010.

[3] A. Adams and M. A. Sasse. Users are not the enemy. *Communications of the ACM*, 42 (12), pages 40–46. ACM, 1999.

[4] D. Adams. *So Long, and Thanks for All the Fish*. Random House, 2008.

[5] A. Adelsbach, S. Gajek, and J. Schwenk. Visual spoofing of SSL protected web sites and effective countermeasures. In *Information Security Practice and Experience*, ISPEC '05, pages 204–217. Springer, 2005.

[6] B. Adida. Beamauth: two-factor web authentication with a bookmark. In *Conference on Computer and Communications Security*, CCS '07, pages 48–57. ACM, 2007.

[7] S. Afroz and R. Greenstadt. Phishzoo: An automated web phishing detection approach based on profiling and fuzzy matching. Technical report, Drexel University, 2009.

[8] S. Afroz and R. Greenstadt. PhishZoo: detecting phishing websites by looking at them. In *Conference on Semantic Computing*, ICSC 2011, pages 368–375. IEEE, 2011.

[9] Alexa Internet. About. URL: `http://www.alexa.com/company` [Online; accessed 2013-05-17].

[10] Amazon Inc. Amazon mechanical turk - welcome. URL: `https://www.mturk.com/mturk/welcome` [Online; accessed 2013-05-02].

[11] 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), pages 1–26. American Accounting Association, 2007.

[12] C. Amrutkar, P. Traynor, and P. C. van Oorschot. An empirical evaluation of security indicators in mobile web browsers. Technical report, Georgia Institute of Technology, 2011.

[13] D. Andreasky. *Enhancing Web Browser Privacy and Security Awareness*. Diploma thesis, University of Munich (LMU), 2010.

[14] Anti-Phishing Working Group (APWG). Phishing acitivity trends report for the month of february 2007. Technical report, Anti-Phishing Working Group (APWG), 2007.

[15] Anti-Phishing Working Group (APWG). APWG report - 1st quarter 2010. Technical report, Anti-Phishing Working Group (APWG), 2010.

[16] Apple Inc. iOS human interface guidelines: Introduction. 2013. URL: `https://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html` [Online; accessed 2013-04-17].

[17] F. Arshad and R. W. Reeder. When usert studies attack: Evaluating security by intentially attacking users. In *Symposium on Usable Privacy and Security (Conference Report)*, SOUPS '05, 16 pages. ACM, 2005.

[18] R. Atterer, M. Wnuk, and A. Schmidt. Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In *Conference on World Wide Web*, WWW '06, pages 203–212, 2006.

[19] AVG Technologies AU Pty Ltd. What is phishing? 2012. URL: `http://resources.avg.com.au/spam/what-is-phishing/` [Online; accessed 2013-03-14].

[20] S. N. Bannur, L. K. Saul, and S. Savage. Judging a site by its content: learning the textual, structural, and visual features of malicious web pages. In *Workshop on Artificial Intelligence and Security*, AISec '11, pages 1–10. ACM, 2011.

[21] J. Bardzell, E. Belvis, and Y.-K. Lim. Human-centered design considerations. In M. Jakobsson and S. Myers, editors, *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*, pages 241–276. Wiley, 2006.

[22] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar. Can machine learning be secure? In *Symposium on Information, Computer and Communications Security*, ASIACCS '06, pages 16–25. ACM, 2006.

[23] S. Bartsch, M. Volkamer, H. Theuerling, and F. Karayumak. Contextualized web warnings, and how they cause distrust. In *Conference on Trust & Trustworthy Computing*, TRUST '13, pages 205–222. Springer, 2013.

[24] A. Beautement and M. A. Sasse. Gathering realistic authentication performance data through field trials. In *Symposium on Usable Privacy and Security*, SOUPS '10, 5 pages. ACM, 2010.

[25] V. Bellotti and A. Sellen. Design for privacy in ubiquitous computing environments. In *European Conference on Computer-Supported Cooperative Work*, ECSCW '93, pages 77–92. Springer, 1993.

[26] C. Benson, A. Elman, S. Nickell, and C. Z. Robertson. GNOME human interface guidelines 2.2.3. 2012. URL: `https://developer.gnome.org/hig-book/stable/` [Online; accessed 2013-04-17].

[27] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986 (INTERNET STANDARD). IETF, 2005. URL: `http://www.ietf.org/rfc/rfc3986.txt`.

[28] T. Berners-Lee, L. Masinter, and M. McCahill. Uniform Resource Locators (URL). RFC 1738 (Proposed Standard). IETF, 1994. URL: `http://www.ietf.org/rfc/rfc1738.txt`.

[29] 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 *Workshop on Cloud Computing Security*, pages 19–30. ACM, 2009.

[30] Binational Working Group on Cross-Border Mass Marketing Fraud. Report on phishing: A report to the minister of public safety and emergency preparedness canada and the attorney general of the united states. Technical report, Binational Working Group on Cross-Border Mass Marketing Fraud, 2006.

[31] J. P. Bliss and C. K. Fallon. Active warnings: False alarms. In M. S. Wogalter, editor, *Handbook of Warnings (Human Factors/Ergonomics)*, pages 231–242. Lawrence Erlbaum Associates, 2006.

[32] A. Blum, B. Wardman, T. Solorio, and G. Warner. Lexical feature based phishing URL detection using online learning. In *Workshop on Artificial Intelligence and Security*, AISec '10, pages 54–60. ACM, 2010.

[33] M. Blythe, H. Petrie, and J. A. Clark. F for fake: Four studies on how we fall for phish. In *Conference on Human Factors in Computing Systems*, CHI '11, pages 3469–3478. ACM, 2011.

[34] C. Bravo-Lillo, L. Cranor, J. Downs, and S. Komanduri. Bridging the gap in computer security warnings: a mental model approach. *Security & Privacy*, 9 (2), pages 18–26. IEEE, 2011.

[35] C. Bravo-Lillo, L. Cranor, J. Downs, S. Komanduri, and M. Sleeper. Improving computer security dialogs. In *Human-Computer Interaction – INTERACT 2011*, pages 18–35. Springer, 2011.

[36] J. C. Brustoloni and R. Villamarín-Salomón. Improving security decisions with polymorphic and audited dialogs. In *Symposium on Usable Privacy and Security*, SOUPS ’07, pages 76–85. ACM, 2007.

[37] Bundesamt für Sicherheit in der Informationstechnik. PCs unter microsoft windows - für privatanwender -. 2012. URL: `https://www.bsi.bund.de/ContentBSI/Themen/Cyber-Sicherheit/Empfehlungen/produktkonfiguration/BSI-E-CS-001.html` [Online; accessed 2012-02-09].

[38] Bundeskriminalamt (BKA). Cybercrime bundeslagebild 2011. Technical report, Bundeskriminalamt, 2012.

[39] CA/Browser Forum. EV SSL certificates. URL: `https://cabforum.org/certificates.html` [Online; accessed 2013-03-14].

[40] P. Cairns. HCI... not as it should be: inferential statistics in HCI research. In *British HCI Group Annual Conference on People and Computers*, BCS HCI ’09, pages 195–201. British Computer Society, 2007.

[41] Y. Cao, W. Han, and Y. Le. Anti-phishing based on automated individual white-list. In *Workshop on Digital Identity Management*, DIM ’08, pages 51–60. ACM, 2008.

[42] M. Chandrasekaran, R. Chinchani, and S. Upadhyaya. Phoney: Mimicking user response to detect phishing attacks. In *Symposium on a World of Wireless, Mobile and Multimedia Networks*, WoWMoM ’06, pages 668–672. IEEE, 2006.

[43] S. A. Chatzichristofis, Y. S. Boutalis, and M. Lux. Selection of the proper compact composite descriptor for improving content based image retrieval. In *Conference on Signal Processing, Pattern Recognition and Applications*, SPPRA ’09. ACTA, 2009.

[44] S. A. Chatzichristofis, K. Zagoris, Y. S. Boutalis, and N. Papamarkos. Accurate image retrieval based on compact composite descriptors and relevance feedback information. *Pattern Recognitionand Artificial Intelligence*, 24 (2), 207 pages. World Scientific Publishing, 2010.

[45] K. T. Chen, J. Y. Chen, C. R. Huang, and C. S. Chen. Fighting phishing with discriminative keypoint features. *Internet Computing*, 13 (3), pages 56–63. IEEE, 2009.

[46] M. C. Chen, J. R. Anderson, and M. H. Sohn. What can a mouse cursor tell us more?: correlation of eye/mouse movements on web browsing. In *Conference on Human Factors in Computing Systems (Extended Abstracts)*, CHI EA ’01, pages 281–282. ACM, 2001.

[47] T.-C. Chen, S. Dick, and J. Miller. Detecting visually similar web pages. *Transactions on Internet Technology*, 10, pages 1–38. ACM, 2010.

[48] S. Chiasson, P. C. Van Oorschot, and R. Biddle. A usability study and critique of two password managers. In *USENIX Security Symposium*, 16 pages. USENIX Association, 2006.

[49] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell. Client-side defense against web-based identity theft. In *Network and Distributed System Security Symposium*, NDSS '04, pages 6:1–6:14. Internet Society, 2004.

[50] L. Cieplinski. MPEG-7 color descriptors and their applications. In *Conference on Computer Analysis of Images and Patterns*, CAIP '01, pages 11–20. Springer, 2001.

[51] Cisco Systems. What is the difference: Viruses, worms, trojans, and bots? 2009. URL: `http://www.cisco.com/web/about/security/intelligence/virus-worm-diffs.html#7` [Online; accessed 2013-03-14].

[52] R. Clayton. Insecure real-world authentication protocols (or why phishing is so profitable). In *Workshop on Security Protocols*, pages 82–88. Springer, 2007.

[53] T. Close. Web security experience, indicators and trust: Scope and use cases. W3C, 2008.

[54] A. Cooper. *About face 2.0: the essentials of interaction design*. Wiley, 2003.

[55] C. Corley. 'Phishing' experiment attracts national adebate about ethics of study, *idsnews.com.* 2005.

[56] A. Costello. Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA). RFC 3492 (Proposed Standard). IETF, 2003. URL: `http://www.ietf.org/rfc/rfc3492.txt`.

[57] M. Cova, C. Kruegel, and G. Vigna. There is no free phish: an analysis of "free" and live phishing kits. In *USENIX Workshop on Offensive Technologies*, WOOT '08, pages 4:1–4:8. USENIX Association, 2008.

[58] N. Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24 (1), pages 87–114. 2001.

[59] L. F. Cranor. What do they indicate?: evaluating security and privacy indicators. *Interactions*, 13 (3), pages 45–47. ACM, 2006.

[60] L. F. Cranor. A framework for reasoning about the human in the loop. In *Workshop on Usability, Psychology, and Security*, 19 pages. USENIX Association, 2008.

[61] L. F. Cranor, P. Guduru, and M. Arjula. User interfaces for privacy agents. *Transactions on Computer-Human Interaction*, 13 (2), pages 135–178. ACM, 2006.

[62] D. Danchev. DIY phishing kits introducing new features | ZDNet. 2008. URL: http://www.zdnet.com/blog/security/diy-phishing-kits-introducing-new-features/1104 [Online; accessed 2012-07-30].

[63] J. Dawes. Do data characteristics change according to the number of scale points used. *Market Research*, 50 (1), pages 61–104. The Market Research Society, 2008.

[64] A. De Luca, B. Frauendienst, M.-E. Maurer, J. Seifert, D. Hausen, N. Kammerer, and H. Hussmann. Does MoodyBoard make internet use more secure?: evaluating an ambient security visualization tool. In *Conference on Human factors in computing systems*, CHI '11, pages 887–890. ACM, 2011.

[65] R. Dhamija and J. Tygar. Phish and hips: Human interactive proofs to detect phishing attacks. In *Workshop on Human Interactive Proofs*, HIP '05, pages 69–83. Springer, 2005.

[66] R. Dhamija and J. D. Tygar. The battle against phishing: Dynamic security skins. In *Symposium on Usable Privacy and Security*, SOUPS '05, pages 77–88. ACM, 2005.

[67] R. Dhamija, J. D. Tygar, and M. Hearst. Why phishing works. In *Conference on Human Factors in Computing Systems*, CHI '06, pages 581–590. ACM, 2006.

[68] digicert Inc. Just how strong is 2048-bit SSL certificate encryption? URL: http://www.digicert.com/TimeTravel/math.htm [Online; accessed 2013-07-21].

[69] P. DiGioia and P. Dourish. Social navigation as a model for usable security. In *Symposium on Usable Privacy and Security*, SOUPS '05, pages 101–108. ACM, 2005.

[70] X. Dong, J. A. Clark, and J. Jacob. Modelling user-phishing interaction. In *Conference on Human System Interactions*, HSI '08, pages 627–632. IEEE, 2008.

[71] P. Dourish, R. E. Grinter, J. Delgado de la Flor, and M. Joseph. Security in the wild: user strategies for managing security as an everyday, practical problem. *Personal and Ubiquitous Computing*, 8 (6), pages 391–401. Springer, 2004.

[72] J. S. Downs, M. Holbrook, and L. F. Cranor. Behavioral response to phishing risk. In *eCrime Researchers Summit*, eCrime '07, pages 37–44. ACM, 2007.

[73] J. S. Downs, M. B. Holbrook, and L. F. Cranor. Decision strategies and susceptibility to phishing. In *Symposium on Usable Privacy and Security*, SOUPS '06, pages 79–90. ACM, 2006.

[74] J. S. Downs, M. B. Holbrook, S. Sheng, and L. F. Cranor. Are your participants gaming the system? In *Conference on Human Factors in Computing Systems*, CHI '10, pages 2399–2402. ACM, 2010.

[75] C. E. Drake, J. J. Oliver, and E. J. Koontz. Anatomy of a phishing email. In *Conference on Email and Anti-Spam*, 8 pages, 2004.

[76] M. Dunlop, S. Groat, and D. Shelly. GoldPhish: using images for content-based phishing analysis. In *Conference on Internet Monitoring and Protection*, ICIMP '12, pages 123–128. IARIA, 2010.

[77] S. Egelman. *Trust Me: Design Patterns for Constructing Trustworthy Trust Indicators*. PhD thesis, Carnegie Mellon University, 2009.

[78] S. Egelman, L. F. Cranor, and J. Hong. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In *Conference on Human Factors in Computing Systems*, CHI '08, pages 1065–1074. ACM, 2008.

[79] S. Egelman, J. King, R. C. Miller, N. Ragouzis, and E. Shehan. Security user studies: methodologies and best practices. In *Conference on Human Factors in Computing Systems (Extended Abstracts)*, CHI EA '07, pages 2833–2836. ACM, 2007.

[80] A. Emigh. The crimeware landscape: Malware, phishing, identity theft and beyond. Technical report, US Department of Homeland Security, SRI International Identity Theft Technology Council, Anti-Phishing Working Group (APWG), 2006.

[81] A. Emigh. Phishing attacks: Information flow and chokepoints. In M. Jakobsson and S. Myers, editors, *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*, pages 31–63. Wiley, 2006.

[82] J. P. Erkkilä. Why we fall for phishing. Technical report, Aalto University, 2011.

[83] P. Faltstrom, P. Hoffman, and A. Costello. Internationalizing Domain Names in Applications (IDNA). RFC 3490 (Proposed Standard). IETF, 2003. URL: `http://www.ietf.org/rfc/rfc3490.txt`.

[84] M. Felegyhazi, C. Kreibich, and V. Paxson. On the potential of proactive domain blacklisting. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats*, LEET '10, 8 pages. USENIX Association, 2010.

[85] E. W. Felten, D. Balfanz, D. Dean, and D. S. Wallach. Web spoofing: An internet con game. In *National Information Systems Security Conference*, NISSC '97, pages 95–103. National Institute of Standards and Technology, 1997.

[86] A. Field. *Discovering Statistics using SPSS: Second Edition*. SAGE Publications Ltd, 2005.

[87] A. P. Field and G. Hole. *How to design and report experiments*. SAGE Publications Ltd, 2003.

[88] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard). IETF, 1999. URL: `http://www.ietf.org/rfc/rfc2616.txt`.

[89] P. Finn and M. Jakobsson. Designing and conducting phishing experiments. *Technology and Society Magazine*, 26 (1), pages 46–58. IEEE, 2007.

[90] R. Fishkin. 14 popular browser toolbars reviewed - the worthwhile and the worthless - moz. 2008. URL: `http://moz.com/blog/12-popular-browser-toolbars-reviewed-the-worthwhile-and-the-worthless` [Online; accessed 2013-06-10].

[91] D. Florencio and C. Herley. Stopping a phishing attack, even when the victims ignore warnings. Technical report, Microsoft, 2005.

[92] D. Florencio and C. Herley. A large-scale study of web password habits. In *Conference on World Wide Web*, WWW '07, pages 657–666. ACM, 2007.

[93] D. Florencio and C. Herley. Is everything we know about password-stealing wrong? *Security & Privacy*, 10 (6), 7 pages. IEEE, 2012.

[94] D. Florêncio and C. Herley. Password rescue: A new approach to phishing prevention. In *USENIX Workshop on Hot Topics in Security*, HOTSEC '06, pages 2:1–2:5. USENIX Association, 2006.

[95] B. J. Fogg, C. Soohoo, D. R. Danielson, L. Marable, J. Stanford, and E. R. Tauber. How do users evaluate the credibility of web sites? a study with over 2,500 participants. In *Conference on Designing for User Experiences*, DUX '03, pages 1–15. ACM, 2003.

[96] J. Friedberg. Internet fraud battlefield. Technical report, Microsoft, 2006.

[97] B. Friedman, D. C. Howe, and E. Felten. Informed consent in the mozilla browser: Implementing value-sensitive design. In *Hawaii International Conference on System Sciences*, HICSS '02, 10 pages. IEEE, 2002.

[98] B. Friedman, D. Hurley, D. C. Howe, E. Felten, and H. Nissenbaum. Users' conceptions of web security: a comparative study. In *Conference on Human Factors in Computing Systems (Extended Abstracts)*, CHI EA '02, pages 746–747. ACM, 2002.

[99] B. Friedman, D. Hurley, D. C. Howe, H. Nissenbaum, and E. Felten. Users' conceptions of risks and harms on the web. In *Conference on Human Factors in Computing Systems (Extended Abstracts)*, CHI EA '02, pages 614–615. ACM, 2002.

[100] A. Fu, X. Deng, and W. Liu. A potential IRI based phishing strategy. In *Conference on Web Information Systems Engineering*, WISE '05, pages 618–619. Springer, 2005.

[101] A. Fu, L. Wenyin, and X. Deng. Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD). *Transactions on Dependable and Secure Computing*, 3 (4), pages 301–311. IEEE, 2006.

[102] S. Furnell, P. Bryant, and A. Phippen. Assessing the security perceptions of personal internet users. *Computers & Security*, 26 (5), pages 410–417. Elsevier, 2007.

[103] E. Gabrilovich and A. Gontmakher. The homograph attack. *Communications of the ACM*, 45 (2), pages 128–128. ACM, 2002.

[104] S. Garera, N. Provos, M. Chew, and A. D. Rubin. A framework for detection and measurement of phishing attacks. In *Workshop on Recurring Malcode*, WORM '07, pages 1:1–1:8. ACM, 2007.

[105] Gartner Research. Gartner says number of phishing e-mails sent to U.S. adults nearly doubles in just two years. 2006. URL: `http://www.gartner.com/newsroom/id/498245` [Online; accessed 2013-02-11].

[106] S. Gibson and L. Laporte. Security now! #277 transcript. 2010. URL: `https://www.grc.com/sn/sn-277.txt` [Online; accessed 2013-07-04].

[107] P. K. Gkonis, C. Z. Patrikakis, A. G. Anadiotis, D. I. Kaklamani, M. T. Andrade, A. Detti, G. Tropea, and N. B. Melazzi. A content-centric, publish-subscribe architecture delivering mobile context-aware health services. In *Future Network & Mobile Summit*, FutureNetw '11, 9 pages. IEEE, 2011.

[108] J. Goecks, W. K. Edwards, and E. D. Mynatt. Challenges in supporting end-user privacy and security management with social navigation. In *Symposium on Usable Privacy and Security*, SOUPS '09, 12 pages. ACM, 2009.

[109] C. Gomes, M. Sellmann, C. Es, and H. Es. The challenge of generating spatially balanced scientific experiment designs. In J.-C. Régin and M. Rueher, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 387–394. Springer, 2004.

[110] C. Gomes, M. Sellmann, C. Van Es, and H. Van Es. Computational methods for the generation of spatially balanced latin squares. URL: `http://www.cs.cornell.edu/gomes/SBLS.htm` [Online; accessed 2010-11-30].

[111] Google Inc. Google safe browsing. URL: `http://www.google.com/tools/firefox/safebrowsing/` [Online; accessed 2013-05-02].

[112] Google Inc. Safe browsing – transparency report – google. 2013. URL: `http://www.google.com/transparencyreport/safebrowsing/?hl=en-US` [Online; accessed 2013-07-25].

[113] S. Görling. The myth of user education. In *Virus Bulletin Conference*, VB '06, 4 pages. Virus Bulletin Ltd, 2006.

[114] A. Gundermann. *Creating a Web Browser for User Studies on Security*. Bachelor thesis, University of Munich (LMU), 2012.

[115] M. Gupta. Spoofing and countermeasures. In M. Jakobsson and S. Myers, editors, *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*, pages 65–104. Wiley, 2006.

[116] D. Gusfield. *Algorithms on strings, trees and sequences: computer science and computational biology*. Cambridge University Press, 1997.

[117] hackers.org. Phishing social networking sites. 2007. URL: `http://ha.ckers.org/blog/20070508/phishing-social-networking-sites/` [Online; accessed 2013-06-17].

[118] D. Harriman. Password fishing on public terminals. *Computer Fraud & Security Bulletin*, 1990 (1), pages 12–14. Elsevier, 1990.

[119] S. Hartman. Requirements for web authentication resistant to phishing. IETF, 2008.

[120] S. Hegt. *Analysis of Current and Future Phishing Attacks on Internet Banking Services*. Master thesis, Technische Universiteit Eindhoven, 2008. URL: `http://alexandria.tue.nl/extra2/afstversl/wsk-i/hegt2008.pdf` [Online; accessed 2013-02-11].

[121] E. Hellier, D. B. Wright, J. Edworthy, and S. Newstead. On the stability of the arousal strength of warning signal words. *Applied Cognitive Psychology*, 14 (6), pages 577–592. Wiley, 2000.

[122] N. Henze, B. Poppinga, and S. Boll. Experiments in the wild: public evaluation of off-screen visualizations in the android market. In *Nordic Conference on Human-Computer Interaction*, NordiCHI ’10, 675–678 pages. ACM, 2010.

[123] C. Herley. So long, and no thanks for the externalities: the rational rejection of security advice by users. In *Workshop on New security Paradigms*, NSPW ’09, pages 133–144. ACM, 2009.

[124] C. Herley. Why do nigerian scammers say they are from nigeria? In *Workshop on the Economics of Information Security*, WEIS ’12, 14 pages, 2012.

[125] C. Herley and D. Florêncio. A profitless endeavor: phishing as tragedy of the commons. In *Workshop on New Security Paradigms*, NSPW ’08, pages 59–70. ACM, 2008.

[126] A. Herzberg and A. Jbara. TrustBar: protecting (even naïve) web users from spoofing and phishing attacks. Technical report, Bar Ilan University, 2004.

[127] A. Herzberg and A. Jbara. Security and identification indicators for browsers against spoofing and phishing attacks. *Transactions on Internet Technology*, 8 (4), pages 16:1–16:36. ACM, 2008.

[128] D. Herzner. *Using Visual Image Comparison to Detect Fraudulent Websites*. Bachelor thesis, University of Munich (LMU), 2012.

[129] L. Höfer. *Bulding a Toolkit for Aggregation and Analyzation of Malicious Web Content with Focus on URL Checking*. Bachelor thesis, University of Munich (LMU), 2011.

[130] C. Y. Huang, S. P. Ma, W. L. Yeh, C. Y. Lin, and C. T. Liu. Mitigate web phishing using site signatures. In *Region 10 Conference*, TENCON '10, pages 803–808. IEEE, 2010.

[131] J. Huang, S. R. Kumar, M. Mitra, W. J. Zhu, and R. Zabih. Image indexing using color correlograms. In *Conference on Computer Vision and Pattern Recognition*, CVPR '97, pages 762–768. IEEE, 1997.

[132] Internet Assigned Numbers Authority. Hypertext transfer protocol (HTTP) status code registry. IANA, 2012. URL: `http://www.iana.org/assignments/http-status-codes/http-status-codes.xml` [Online; accessed 2013-05-21].

[133] T. Jagatic and M. Jakobsson. Phishing attacks using social networks. 2005. URL: `http://www.indiana.edu/~phishing/social-network-experiment/`.

[134] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Communications of the ACM*, 50 (10), pages 94–100. ACM, 2007.

[135] M. Jakobsson. Modeling and preventing phishing attacks. In *Conference on Financial Cryptography and Data Security*, pages 89–89. Springer, 2005.

[136] M. Jakobsson. The human factor in phishing. Technical report, Indiana University, 2007.

[137] M. Jakobsson and S. Myers. *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*. Wiley, 2006.

[138] M. Jakobsson and S. Myers. Delayed password disclosure. *SIGACT News*, 38 (3), pages 47–59. ACM, 2007.

[139] M. Jakobsson, A. Tsow, A. Shah, E. Blevis, and Y.-K. Lim. What instills trust? a qualitative study of phishing. In *Conference on Financial Cryptography and Data Security*, pages 356–361. Springer, 2007.

[140] M. Jakobsson and A. Young. Distributed phishing attacks. In *DIMACS Workshop on Theft in E-Commerce*, 10 pages, 2005.

[141] Javelin Strategy & Research Inc. 2013 identity fraud report: Data breaches becoming a treasure trove for fraudsters. Technical report, Javelin Strategy & Research Inc., 2013.

[142] L. Jiao, S. H. Lim, N. Bhatti, Y. Xiong, and J. Liu. Style and branding elements extraction from businessweb sites. In *Symposium on Document Engineering*, DocEng '10, pages 231–234. ACM, 2010.

[143] J. Johnston, J. Eloff, and L. Labuschagne. Security and human computer interfaces. *Computers & Security*, 22 (8), pages 675–684. Elsevier, 2003.

[144] J. Jung and E. Sit. An empirical study of spam traffic and the use of DNS black lists. In *Conference on Internet Measurement*, IMC '04, pages 370–375. ACM, 2004.

[145] C. Karlof, J. D. Tygar, and D. Wagner. Conditioned-safe ceremonies and a user study of an application to web authentication. In *Symposium on Usable Privacy and Security*, SOUPS '09, 20 pages. ACM, 2009.

[146] Kaspersky Lab ZAO. Kaspersky lab report: 37.3 million users experienced phishing attacks in the last year. Technical report, Kaspersky Lab ZAO, 2013.

[147] R. Kay. QuickStudy: phishing, *Computerworld*. 2004. URL: `http://www.computerworld.com/s/article/89096/Phishing` [Online; accessed 2013-02-13].

[148] G. Keizer. Phishing costs nearly $1 billion, *Informationweek*. 2005. URL: `http://www.informationweek.com/news/164902704` [Online; accessed 2013-02-11].

[149] G. Keizer. Phishers beat bank's two-factor authentication, *Informationweek*. 2006. URL: `http://www.informationweek.com/news/190400362#` [Online; accessed 2012-07-30].

[150] S. Kempe. *Enhancing Datatype Based Security Notifications For Websites*. Bachelor thesis, University of Munich (LMU), 2011.

[151] F. D. Keukelaere, S. Yoshihama, S. Trent, Y. Zhang, L. Luo, and M. E. Zurko. Adaptive security dialogs for improved security behavior of users. In *Human-Computer Interaction – INTERACT 2009*, Lecture Notes in Computer Science, pages 510–523. Springer, 2009.

[152] E. Kirda and C. Kruegel. Protecting users against phishing attacks with AntiPhish. In *Computer Software and Applications Conference*, pages 517–524. IEEE, 2005.

[153] P. Kolari, T. Finin, and A. Joshi. SVMs for the blogosphere: Blog identification and splog detection. In *Symposium on Computational Approaches to Analyzing Weblogs*, pages 92–100. AAAI, 2006.

[154] V. Krammer. Phishing defense against IDN address spoofing attacks. In *Conference on Privacy, Security and Trust*, PST '06, pages 32:1–32:9. ACM, 2006.

[155] P. Kumaraguru, Y. Rhee, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge. Protecting people from phishing: the design and evaluation of an embedded training email system. In *Conference on Human Factors in Computing Systems*, CHI '07, pages 905–914. ACM, 2007.

[156] P. Kumaraguru, Y. Rhee, S. Sheng, S. Hasan, A. Acquisti, L. F. Cranor, and J. Hong. Getting users to pay attention to anti-phishing education: evaluation of retention and transfer. In *eCrime Researchers Summit*, eCrime '07, pages 70–81. ACM, 2007.

[157] P. Kumaraguru, S. Sheng, A. Acquisti, L. F. Cranor, and J. Hong. Teaching johnny not to fall for phish. *Transactions on Internet Technology*, 10, pages 7:1–7:31. ACM, 2010.

[158] I. Lam, W. Xiao, S. Wang, and K. Chen. Counteracting phishing page polymorphism: An image layout analysis approach. In *Conference on Advances in Information Security and Assurance*, ISA '09, pages 270–279. Springer, 2009.

[159] P. Le Hégaret, R. Whitmer, and L. Wood. Document object model. W3C, 2009. URL: `http://www.w3.org/DOM/` [Online; accessed 2013-05-15].

[160] M. Levene. *An Introduction to Search Engines and Web Navigation*. Wiley, 2011.

[161] J. R. Lewis. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *Human-Computer Interaction*, 7 (1), pages 57–78. L. Erlbaum Associates Inc., 1995.

[162] J. Leyden. Phishing losses overestimated - survey • the register. 2004. URL: `http://www.theregister.co.uk/2004/12/03/phishing_survey_towergroup/` [Online; accessed 2013-02-11].

[163] J. Leyden. US phishing losses hit $500m • the register. 2004. URL: `http://web4.theregister.co.uk/2004/09/29/phishing_survey/` [Online; accessed 2013-02-11].

[164] M. Li, X. Chen, X. Li, B. Ma, and P. Vitanyi. The similarity metric. *Transactions on Information Theory*, 50 (12), pages 3250–3264. IEEE, 2004.

[165] M. Liberman. Language log: Phishing. 2004. URL: `http://itre.cis.upenn.edu/~myl/languagelog/archives/001477.html` [Online; accessed 2013-02-26].

[166] E. Lin, S. Greenberg, E. Trotter, D. Ma, and J. Aycock. Does domain highlighting help people identify phishing sites? In *Conference on Human Factors in Computing Systems*, CHI '11, pages 2075–2084. ACM, 2011.

[167] A. Litan. Phishing attack victims likely targets for identity theft. Technical report, Gartner Research, 2004.

[168] W. Liu, X. Deng, G. Huang, and A. Fu. An antiphishing strategy based on visual similarity assessment. *Internet Computing*, 10 (2), pages 58–65. IEEE, 2006.

[169] W. Liu, G. Huang, L. Xiaoyue, X. Deng, and Z. Min. Phishing web page detection. In *Conference on Document Analysis and Recognition*, ICDAR '05, pages 560–564. IEEE, 2005.

[170] K. Lodrick. Anti-malware expert and CEO, eugene kaspersky, offers theory for stopping cybercrime, *Examiner.com*. 2009. URL: `http://www.examiner.com/article/anti-malware-expert-and-ceo-eugene-kaspersky-offers-theory-for-stopping-cyber` [Online; accessed 2012-08-14].

[171] M. Lux and S. Chatzichristofis. Lire: lucene image retrieval: an extensible java CBIR library. In *Conference on Multimedia*, CMS '08, pages 1085–1088. Springer, 2008.

[172] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In *Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 1245–1254. ACM, 2009.

[173] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Identifying suspicious URLs: an application of large-scale online learning. In *Conference on Machine Learning*, ICML '09, pages 681–688. ACM, 2009.

[174] N. A. Macmillan and C. D. Creelman. *Detection Theory: A User's Guide*. Psychology Press, 2004.

[175] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *Transactions on Pattern Analysis and Machine Intelligence*, 11 (7), pages 674–693. IEEE, 1989.

[176] M. Mannan and P. C. van Oorschot. Security and usability: the gap in real-world online banking. In *Workshop on New Security Paradigms*, NSPW '07, pages 1–14. ACM, 2008.

[177] G. Markham. Phishing - browser-based defences. 2005. URL: `http://www.gerv.net/security/phishing-browser-defences.html` [Online; accessed 2012-08-06].

[178] MarkMonitor. Brandjacking index - spring 2009. Technical report, MarkMonitor, 2009.

[179] MarkMonitor. Brandjacking index - 2009 - the year in review. Technical report, MarkMonitor, 2010.

[180] J. M. Martínez. MPEG-7 overview. The Moving Picture Experts Group, 2004. URL: http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm#E12E29 [Online; accessed 2012-09-16].

[181] M.-E. Maurer. Poster: Community-based security and privacy protection during web browsing. In *Symposium on Usable Privacy and Security*, SOUPS '10, 2 pages. ACM, 2010.

[182] M.-E. Maurer, A. De Luca, and H. Hussmann. Data type based security alert dialogs. In *Conference on Human factors in computing systems (Extended Abstracts)*, CHI EA '11, 2359–2364 pages. ACM, 2011.

[183] M.-E. Maurer, A. De Luca, and S. Kempe. Using data type based security alert dialogs to raise online security awareness. In *Symposium on Usable Privacy and Security*, SOUPS '11, pages 2:1–2:13. ACM, 2011.

[184] M.-E. Maurer, A. De Luca, and T. Stockinger. Shining chrome: using web browser personas to enhance SSL certificate visualization. In *Human-Computer Interaction–INTERACT 2011*, pages 44–51. Springer, 2011.

[185] M.-E. Maurer and D. Herzner. Using visual website similarity for phishing detection and reporting. In *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts*, 1625–1630 pages, 2012.

[186] M.-E. Maurer and L. Höfer. Sophisticated phishers make more spelling mistakes: using URL similarity against phishing. In *Cyberspace Safety and Security*, pages 414–426. Springer, 2012.

[187] P. Mavrommatis and N. Provos. Google online security blog: Introducing google's online security efforts. 2007. URL: http://googleonlinesecurity.blogspot.de/2007/05/introducing-googles-anti-malware.html [Online; accessed 2013-02-26].

[188] McAfee. Rootkits, part 1 of 3: The growing threat. Technical report, McAfee, 2006.

[189] D. K. McGrath and M. Gupta. Behind phishing: an examination of phisher modi operandi. In *USENIX Workshop on Large-Scale Exploits and Emergent Threats*, LEET'08, pages 4:1–4:8. USENIX Association, 2008.

[190] R. McMillan. Gartner: Consumers to lose $2.8b to phishers in 2006, *Computerworld*. 2006. URL: http://www.computerworld.com/s/article/9004926/Gartner_Consumers_to_lose_2.8B_to_phishers_in_2006 [Online; accessed 2013-02-11].

[191] E. Medvet, E. Kirda, and C. Kruegel. Visual-similarity-based phishing detection. In *Conference on Security and Privacy in Communication Networks*, SecureComm '08, 22:1–22:6 pages. ACM, 2008.

[192] M. Mühlbauer. *User Interfaces for Indication of Visual Website Similarity for Fraudulent Websites*. Master thesis, University of Munich (LMU), 2012.

[193] Microsoft. Error messages. URL: `http://msdn.microsoft.com/en-us/library/windows/desktop/aa511267.aspx` [Online; accessed 2013-04-17].

[194] Microsoft. SmartScreen filter - microsoft windows, *windows.microsoft.com*. URL: `http://windows.microsoft.com/en-us/internet-explorer/products/ie-9/features/smartscreen-filter` [Online; accessed 2013-05-02].

[195] Microsoft. Windows user experience interaction guidelines. URL: `http://msdn.microsoft.com/en-us/library/windows/desktop/aa511258.aspx` [Online; accessed 2013-04-17].

[196] Microsoft. Windows vista application development requirements for user account control (UAC). 2006. URL: `http://msdn.microsoft.com/en-us/library/aa905330.aspx` [Online; accessed 2013-05-15].

[197] G. A. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63 (2), pages 81–97. American Psychological Association, 1956.

[198] D. Miyamoto, H. Hazeyama, and Y. Kadobayashi. SPS: a simple filtering algorithm to thwart phishing attacks. In *Conference on Technologies for Advanced Heterogeneous Networks*, AINTEC '05, pages 195–209. Springer, 2005.

[199] D. Miyamoto, H. Hazeyama, and Y. Kadobayashi. An evaluation of machine learning-based methods for detection of phishing sites. In M. Köppen, N. Kasabov, and G. Coghill, editors, *Conference on Advances in Neuro-Information Processing*, pages 539–546. Springer, 2009.

[200] mk590. AOL for free? - google groups. 1996. URL: `https://groups.google.com/forum/#!topic/alt.2600/c7k2bykZz5A/discussion` [Online; accessed 2013-02-19].

[201] F. Müller. *Keyword Based Security Awareness Warnings for Websites*. Project thesis, University of Munich (LMU), 2010.

[202] P. Mockapetris. Domain names - concepts and facilities. RFC 1034 (INTERNET STANDARD). IETF, 1987. URL: `http://www.ietf.org/rfc/rfc1034.txt`.

[203] T. Moore and R. Clayton. An empirical analysis of the current state of phishing attack and defence. In *Workshop on the Economics of Information Security*, WEIS '07, 20 pages, 2007.

[204] T. Moore and R. Clayton. Examining the impact of website take-down on phishing. In *eCrime Researchers Summit*, eCrime '07, 1–13 pages. ACM, 2007.

[205] T. Moore and R. Clayton. The consequence of non-cooperation in the fight against phishing. In *eCrime Researchers Summit*, eCrime ’08, 14 pages. IEEE, 2008.

[206] T. Moore and R. Clayton. Evaluating the wisdom of crowds in assessing phishing websites. In *Conference on Financial Cryptography and Data Security*, FC ’08, pages 16–30. Springer, 2008.

[207] T. Moore and R. Clayton. Evil searching: Compromise and recompromise of internet hosts for phishing. In *Conference on Financial Cryptography and Data Security*, FC ’09, pages 256–272. Springer, 2009.

[208] T. Moore, R. Clayton, and H. Stern. Temporal correlations between spam and phishing websites. In *USENIX Workshop on Large-scale Exploits and Emergent Threats*, LEET ’09, 8 pages. USENIX Association, 2009.

[209] B. Morton. SSLPersonas | entrust, inc. 2010. URL: https://www.entrust.com/sslpersonas/ [Online; accessed 2013-07-04].

[210] S. Motiee, K. Hawkey, and K. Beznosov. Do windows users follow the principle of least privilege?: investigating user account control practices. In *Symposium on Usable Privacy and Security*, SOUPS ’10, pages 1:1–1:13. ACM, 2010.

[211] S. Mustaca. Phishing, spam and malware statistics for february 2011 | avira – TechBlog. 2011. URL: http://techblog.avira.com/2011/03/12/phishing-spam-and-malware-statistics-for-february-2011/en/ [Online; accessed 2013-07-08].

[212] S. Myers. Introduction to phishing. In M. Jakobsson and S. Myers, editors, *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*, pages 1–29. Wiley, 2006.

[213] B. Nahorney. Symantec intelligence report: November 2012. Technical report, Symantec Intelligence, 2012.

[214] National Consumers League. About NCL - national consumers league. URL: http://www.nclnet.org/about-ncl [Online; accessed 2013-02-19].

[215] National Consumers League. A call for action against phishing scams. Technical report, National Consumers League, 2006.

[216] J. Nielsen. 10 heuristics for user interface design. 1995. URL: http://www.nngroup.com/articles/ten-usability-heuristics/ [Online; accessed 2013-07-22].

[217] J. Nielsen. User education is not the answer to security problems. 2004. URL: http://www.useit.com/alertbox/20041025.html [Online; accessed 2012-08-06].

[218] D. A. Norman. *The design of everyday things*. Basic Books, 2002.

[219] NSS Labs. 2013 browser security comparative analysis: Socially engineered malware. 2013. URL: https://www.nsslabs.com/reports/2013-browser-security-comparative-analysis-socially-engineered-malware [Online; accessed 2013-07-24].

[220] T. O'Brien. Gone spear-phishin' - new york times. 2005. URL: http://www.nytimes.com/2005/12/04/business/yourmoney/04spear.html?pagewanted=all [Online; accessed 2012-08-01].

[221] G. Ollmann. The phishing guide: Understanding and preventing phishing attacks. Technical report, IBM Internet Security Systems, 2005.

[222] OnGuard Online. Phishing. 2011. URL: http://www.onguardonline.gov/phishing [Online; accessed 2013-04-02].

[223] P. C. v. Oorschot and S. Stubblebine. Countering identity theft through digital uniqueness, location cross-checking, and funneling. In A. S. Patrick and M. Yung, editors, *Conference on Financial Cryptography and Data Security*, FC '05, pages 31–43. Springer, 2005.

[224] L. Opennheimer. Phishing, PayPal and the challenges of reporting accurate data | OpenDNS blog. 2011. URL: http://blog.opendns.com/2011/02/28/phishing-paypal-and-the-challenges-of-reporting-accurate-data/ [Online; accessed 2013-06-03].

[225] R. Oppliger and S. Gajek. Effective protection against phishing and web spoofing. In *Conference on Communications and Multimedia Security*, CMS '05, pages 32–41. Springer, 2005.

[226] V. Palant. URL fixer for mozilla firefox. URL: http://urlfixer.org/ [Online; accessed 2013-04-02].

[227] V. Palant. Adblock plus and (a little) more: Typo correction feature in adblock plus. 2012. URL: http://adblockplus.org/blog/typo-correction-feature-in-adblock-plus [Online; accessed 2013-04-02].

[228] Y. Pan and X. Ding. Anomaly based web phishing page detection. In *Computer Security Applications Conference*, ACSAC '06, pages 381–392. IEEE, 2006.

[229] E. Papachristos and N. Avouris. Are first impressions about websites only related to visual appeal? In P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler, editors, *Human-Computer Interaction – INTERACT 2011*, pages 489–496. Springer, 2011.

[230] B. Parno, C. Kuo, and A. Perrig. Phoolproof phishing prevention. In G. Di Crescenzo and A. Rubin, editors, *Conference on Financial Cryptography and Data Security*, pages 1–19. Springer, 2006.

[231] N. Perlroth. Chinese hackers infiltrate new york times computers, *The New York Times*. 2013. URL: http://www.nytimes.com/2013/01/31/technology/chinese-hackers-infiltrate-new-york-times-computers.html [Online; accessed 2013-02-15].

[232] PhishMe. Spear phishing awareness training. URL: http://www.phishme.com/ [Online; accessed 2013-04-02].

[233] PhishTank. PhishTank > what is phishing? (definition of phishing, with examples). URL: http://www.phishtank.com/what_is_phishing.php?view=website&annotated=true [Online; accessed 2013-02-13].

[234] PhishTank. Statistics about phishing activity and PhishTank usage > january 2013. 2013. URL: http://www.phishtank.com/stats/2013/01/ [Online; accessed 2013-03-14].

[235] Pingdom.com. Internet 2012 in numbers. 2013. URL: http://royal.pingdom.com/2013/01/16/internet-2012-in-numbers/ [Online; accessed 2013-02-11].

[236] D. Platt Majoras, O. Swindle, T. B. Leary, P. Jones Harbour, and J. Leibowitz. Spyware workshop report: Monitoring software on your PC: spyware, adware, and other software. staff report. Technical report, Federal Trade Commission, 2005.

[237] J. Postel. Domain Name System Structure and Delegation. RFC 1591 (Informational). IETF, 1994. URL: http://www.ietf.org/rfc/rfc1591.txt.

[238] P. Prakash, M. Kumar, R. Kompella, and M. Gupta. PhishNet: predictive blacklisting to detect phishing attacks. In *Conference on Computer Communications*, INFOCOM '10, pages 1–5. IEEE, 2010.

[239] N. Provos. Google online security blog: Safe browsing - protecting web users for 5 years and counting. 2012. URL: http://googleonlinesecurity.blogspot.jp/2012/06/safe-browsing-protecting-web-users-for.html [Online; accessed 2013-02-13].

[240] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu. The ghost in the browser analysis of web-based malware. In *Workshop on Hot Topics in Understanding Botnets*, HotBots '07, 4 pages. USENIX Association, 2007.

[241] S. Pruitt. AOL fights phishing, *PCWorld*. 2005. URL: http://www.pcworld.com/article/120512/article.html [Online; accessed 2013-02-19].

[242] T. Raffetseder, E. Kirda, and C. Kruegel. Building anti-phishing browser plug-ins: An experience report. In *Workshop on Software Engineering for Secure Systems*, SESS '07, 7 pages. IEEE, 2007.

[243] R. Rasmussen, G. Aaron, and A. Routt. Global phishing survey: Trends and domain name use in 1H2012. 2012. URL: http://docs.apwg.org/reports/APWG_GlobalPhishingSurvey_1H2012.pdf [Online; accessed 2013-02-11].

[244] C. Reithmeier. *Diminishing Visual Brand Trust on Websites for better Security Assessment*. Bachelor thesis, University of Munich (LMU), 2012.

[245] R. A. Rensink, J. K. O'Regan, and J. J. Clark. To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science*, 8 (5), 368–373 pages. SAGE Publications, 1997.

[246] E. Robertson. A phish tale? moving from hype to reality. Technical report, TowerGroup, 2004.

[247] T. Ronda, S. Saroiu, and A. Wolman. Itrustpage: a user-assisted anti-phishing tool. In *Conference on Computer Systems*, pages 261–272. ACM, 2008.

[248] A. P. Rosiello, E. Kirda, and F. Ferrandi. A layout-similarity-based approach for detecting phishing pages. In *Conference on Security and Privacy in Communications Networks*, SecureComm '07, pages 454–463. IEEE, 2007.

[249] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell. Stronger password authentication using browser extensions. In *USENIX Security Symposium*, 15 pages. USENIX Association, 2005.

[250] J. Ross, L. Irani, M. S. Silberman, A. Zaldivar, and B. Tomlinson. Who are the crowdworkers?: shifting demographics in mechanical turk. In *Conference on Human Factors in Computing Systems (Extended Abstracts)*, CHI EA '10, pages 2863–2872. ACM, 2010.

[251] M. Rouse. What is phishing? - definition from WhatIs.com. 2007. URL: http://searchsecurity.techtarget.com/definition/phishing [Online; accessed 2013-02-13].

[252] RSA Security. Protecting against phishing by implementing strong two-factor authentication. Technical report, RSA Security, 2004.

[253] N. Rubenking. Blocking known and unknown frauds - norton 360 version 6.0 review & rating | PCMag.com. 2012. URL: http://www.pcmag.com/article2/0,2817,2400194,00.asp [Online; accessed 2013-04-02].

[254] S. Sanyal and S. H. Sengamedu. LogoSeeker: a system for detecting and matching logos in natural images. In *Conference on Multimedia*, MULTIMEDIA '07, pages 166–167. ACM, 2007.

[255] M. A. Sasse and I. Flechais. Usable security: Why do we need it? how do we get it? In L. F. Cranor and S. Garfinkel, editors, *Security and Usability: Designing secure systems that people can use*, pages 13–30. O'Reilly, 2005.

[256] A. Sawall. Kaspersky: "Phishing-Angriffe sind nachweislich erfolgreich" - golem.de. 2013. URL: `http://www.golem.de/news/kaspersky-phishing-angriffe-sind-nachweislich-erfolgreich-1306-99933.html` [Online; accessed 2013-07-24].

[257] S. Schechter. Common pitfalls in writing about security and privacy human subjects experiments, and how to avoid them. Technical report, Microsoft, 2010.

[258] S. Schechter and M. Smith. How much security is enough to stop a thief? In *Conference on Financial Cryptography and Data Security*, FC '03, pages 122–137. Springer, 2003.

[259] 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 *Symposium on Security and Privacy*, SP '07, pages 51–65. ACM, 2007.

[260] F. Schneider, N. Provos, R. Moll, M. Chew, and B. Rakowski. Phishing protection: Design documentation - MozillaWiki. 2008. URL: `https://wiki.mozilla.org/Phishing_Protection:_Design_Documentation` [Online; accessed 2012-07-30].

[261] S. Sheng, M. Holbrook, P. Kumaraguru, L. Cranor, and J. Downs. Who falls for phish? a demographic analysis of phishing susceptibility and effectiveness of interventions. In *Conference on Human Factors in Computing Systems*, CHI '10, pages 373–382. ACM, 2010.

[262] S. Sheng, P. Kumaraguru, A. Acquisti, L. Cranor, and J. Hong. Improving phishing countermeasures: An analysis of expert interviews. In *eCrime Researchers Summit*, eCrime '09, pages 1–15. IEEE, 2009.

[263] 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 *Symposium on Usable Privacy and Security*, SOUPS '07, pages 88–99. ACM, 2007.

[264] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang. An empirical analysis of phishing blacklists. In *Conference on Email and Anti-Spam*, CEAS '09, 10 pages, 2009.

[265] D. Shin and R. Lopes. An empirical study of visual security cues to prevent the SSLstripping attack. In *Computer Security Applications Conference*, ACSAC '11, pages 287–296. ACM, 2011.

[266] M. Shin, C. Straub, R. Tamassia, and D. J. Polivy. Authenticating web content with prooflets. Technical report, Brown University, 2002.

[267] D. J. Simons and D. T. Levin. Change blindness. *Trends in Cognitive Sciences*, 1 (7), 261–267 pages. Cell Press, 1997.

[268] T. L. Smith-Jackson and M. S. Wogalter. Methods and procedures in warning research. In M. S. Wogalter, editor, *Handbook of Warnings (Human Factors/Ergonomics)*, pages 23–33. Lawrence Erlbaum Associates, 2006.

[269] J. Sobey. *An evaluation of new browser indicators for Extended Validation certificates*. Master thesis, Carleton University, 2008. URL: `https://www.ccsl.carleton.ca/people/theses/Sobey_Master_Thesis_08.pdf` [Online; accessed 2013-02-11].

[270] J. Sobey, R. Biddle, P. C. Oorschot, and A. S. Patrick. Exploring user reactions to new browser cues for extended validation certificates. In *European Symposium on Research in Computer Security*, ESORICS '08, pages 411–427. Springer, 2008.

[271] A. Sotirakopoulos, K. Hawkey, and K. Beznosov. "I did it because i trusted you": Challenges with the study environment biasing participant behaviours. In *Symposium on Usable Privacy and Security*, SOUPS '10, 6 pages. ACM, 2010.

[272] SSL Security Blog. Firefox 4 gets rid of the padlock icon. 2011. URL: `http://www.whynopadlock.com/blog/2011/03/firefox-4-gets-rid-of-the-padlock-icon/` [Online; accessed 2013-02-26].

[273] F. Stajano and P. Wilson. Understanding scam victims: seven principles for systems security. *Communications of the ACM*, 54 (3), pages 70–75. ACM, 2011.

[274] S. Stamm, Z. Ramzan, and M. Jakobsson. Drive-by pharming. In *Conference on Information and Communications Security*, pages 495–506. Springer, 2007.

[275] Stanford Persuasive Technology Lab. Stanford guidelines for web credibility. Technical report, Stanford Persuasive Technology Lab, 2002.

[276] D. Stebila. Reinforcing bad behaviour: the misuse of security indicators on popular websites. In *Conference of the Computer-Human Interaction Special Interest Group of Australia on Computer-Human Interaction*, OZCHI '10, pages 248–251. ACM, 2010.

[277] M. Stepp and C. Collberg. Browser toolbars. In S. Jakobsson and S. Myers, editors, *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*, pages 493–521. Wiley, 2006.

[278] G. Stewart. AOL cert: lugs-lousy - google groups. 1995. URL: `https://groups.google.com/forum/#!topic/alt.2600/vVIytx-vu9M/discussion` [Online; accessed 2013-02-19].

[279] T. Stockinger. *Enhancing SSL Awareness in Web Browsers*. Master thesis, University of Munich (LMU), 2010.

[280] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying wolf: An empirical study of SSL warning effectiveness. In *USENIX Security Symposium*, 18 pages. USENIX Association, 2009.

[281] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *Transactions on Systems, Man and Cybernetics*, 8 (6), pages 460–473. IEEE, 1978.

[282] Techterms.com. Phishing definition. 2006. URL: `http://www.techterms.com/definition/phishing` [Online; accessed 2013-02-13].

[283] R. Tervo. Secrets of the LUHN-10 algorithm, *Secrets of the LUHN-10 Algorithm*. 2002. URL: `http://www.ee.unb.ca/tervo/ee4253/luhn.shtml` [Online; accessed 2011-03-03].

[284] A. Tsow and M. Jakobsson. Deceit and deception: A large user study of phishing. Technical Report 649, Indiana University, 2007.

[285] TÜV Rheinland. Unsere geschichte. URL: `http://www.tuv.com/de/deutschland/ueber_uns/daten_fakten/geschichte/geschichte_tuev_rheinland.html` [Online; accessed 2013-07-06].

[286] J. Ullrich. ISC diary | MySpace phish and drive-by attack vector propagating fast flux network growth. 2007. URL: `http://isc.sans.edu/diary/MySpace+Phish+and+Drive-by+attack+vector+propagating+Fast+Flux+network+growth/3060` [Online; accessed 2013-02-26].

[287] US-CERT. Avoiding social engineering and phishing attacks. 2009. URL: `http://www.us-cert.gov/ncas/tips/ST04-014` [Online; accessed 2013-04-02].

[288] R. Vamosi. Meet larry, firefox's friendly passport officer, *CNET*. 2008. URL: `http://news.cnet.com/8301-10789_3-9970606-57.html` [Online; accessed 2013-02-11].

[289] W3C. Frames in HTML documents. W3C, 1999. URL: `http://www.w3.org/TR/REC-html40/present/frames.html` [Online; accessed 2013-05-21].

[290] B. Wardman and G. Warner. Automating phishing website identification through deep MD5 matching. In *eCrime Researchers Summit*, eCrime '08, pages 1–7. IEEE, 2008.

[291] R. Wash. Folk models of home computer security. In *Symposium on Usable Privacy and Security*, SOUPS '10, pages 11:1–11:16. ACM, 2010.

[292] D. Watson, T. Holz, and S. Mueller. Know your enemy: Phishing, *The Honeynet Project*. 2005. URL: http://www.honeynet.org/papers/phishing/ [Online; accessed 2013-02-11].

[293] M. Wertheimer and K. Riezler. Gestalt theory. *Social Research*, pages 78–99. The New School, 1944.

[294] T. Whalen and K. M. Inkpen. Gathering evidence: use of visual security cues in web browsers. In *Conference on Graphics Interface*, GI '05, pages 137–144. Canadian Human-Computer Communications Society, 2005.

[295] C. Whittaker, B. Ryner, and M. Nazif. Large-scale automatic classification of phishing pages. In *Network and Distributed System Security Symposium*, NDSS '10, pages 8:1–8:14. Internet Society, 2010.

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

[297] S. Wicha. *Community-Based Security and Privacy Ratings for Internet Websites*. Bachelor thesis, University of Munich (LMU), 2010.

[298] T. D. Wickens. *Elementary Signal Detection Theory*. Oxford University Press, 2001.

[299] Wikipedia. Amazon mechanical turk, *Wikipedia, the free encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=Amazon_Mechanical_Turk&oldid=551309269 [Online; accessed 2013-05-02].

[300] Wikipedia. bzip2, *Wikipedia, the free encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=Bzip2&oldid=539661584 [Online; accessed 2013-04-12].

[301] Wikipedia. Hash function, *Wikipedia, the free encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=Hash_function&oldid=552770457 [Online; accessed 2013-05-02].

[302] Wikipedia. Honeypot (computing), *Wikipedia, the free encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=Honeypot_(computing)&oldid=552758252 [Online; accessed 2013-05-02].

[303] Wikipedia. Host (network), *Wikipedia, the free encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=Host_(network)&oldid=558330185 [Online; accessed 2013-07-25].

[304] Wikipedia. hosts (file), *Wikipedia, the free encyclopedia*. 2013. URL: https://en.wikipedia.org/w/index.php?title=Hosts_(file)&oldid=558409375 [Online; accessed 2013-06-06].

[305] Wikipedia. Lempel–Ziv–Markov chain algorithm, *Wikipedia, the free encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=Lempel%E2%80%93Ziv%E2%80%93Markov_chain_algorithm&oldid=549736394 [Online; accessed 2013-04-12].

[306] Wikipedia. Multi-factor authentication, *Wikipedia, the free encyclopedia*. 2013. URL: https://en.wikipedia.org/w/index.php?title=Multi-factor_authentication&oldid=567301291 [Online; accessed 2013-08-07].

[307] Wikipedia. n-gram, *Wikipedia, the free encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=N-gram&oldid=548024834 [Online; accessed 2013-05-02].

[308] Wikipedia. One-time password, *Wikipedia, the free encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=One-time_password&oldid=551872746 [Online; accessed 2013-05-02].

[309] Wikipedia. PageRank, *Wikipedia, the free encyclopedia*. 2013. URL: https://en.wikipedia.org/w/index.php?title=PageRank&oldid=552323936 [Online; accessed 2013-05-02].

[310] Wikipedia. Pretty good privacy, *Wikipedia, the free encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=Pretty_Good_Privacy&oldid=552980140 [Online; accessed 2013-05-02].

[311] Wikipedia. RSA (security firm), *Wikipedia, the free encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=RSA_(security_firm)&oldid=551867617 [Online; accessed 2013-05-02].

[312] Wikipedia. Time to live, *Wikipedia, the free encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=Time_to_live&oldid=544644245 [Online; accessed 2013-05-02].

[313] Wikipedia. Variable shadowing, *Wikipedia, the free encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=Variable_shadowing&oldid=552379488 [Online; accessed 2013-08-13].

[314] Wikipedia. Whois, *Wikipedia, the free encyclopedia*. 2013. URL: http://en.wikipedia.org/w/index.php?title=Whois&oldid=549772856 [Online; accessed 2013-05-02].

[315] M. S. Wogalter. Communication-human information processing (c-HIP) model. In M. S. Wogalter, editor, *Handbook of Warnings (Human Factors/Ergonomics)*, pages 51–61. Lawrence Erlbaum Associates, 2006.

[316] M. S. Wogalter. *Handbook of Warnings (Human Factors/Ergonomics)*. Lawrence Erlbaum Associates, 2006.

[317] M. S. Wogalter. Purposes and scope of warnings. In M. S. Wogalter, editor, *Handbook of Warnings (Human Factors/Ergonomics)*, pages 3–9. Lawrence Erlbaum Associates, 2006.

[318] M. S. Wogalter, V. C. Conzola, and T. L. Smith-Jackson. Research-based guidelines for warning design and evaluation. *Applied Ergonomics*, 33 (3), pages 219–230. Elsevier, 2002.

[319] M. S. Wogalter, V. C. Conzola, and W. J. Vigilante. Applying usability engineering principles to the design and testing of warning text. In M. S. Wogalter, editor, *Handbook of Warnings (Human Factors/Ergonomics)*, pages 487–498. Lawrence Erlbaum Associates, 2006.

[320] M. S. Wogalter and C. B. Mayhorn. The future of risk communication: Technology-based warning systems. In M. S. Wogalter, editor, *Handbook of Warnings (Human Factors/Ergonomics)*, pages 3–9. Lawrence Erlbaum Associates, 2006.

[321] Wombat Security. Train employees to identify malicious URLs. URL: `http://www.wombatsecurity.com/antiphishingphil` [Online; accessed 2013-04-02].

[322] WOT (Web of Trust). Safe browsing tool WOT (web of trust). URL: `http://www.mywot.com/` [Online; accessed 2013-04-15].

[323] WOT (Web of Trust). Safe browsing tool. 2013. URL: `http://www.mywot.com/` [Online; accessed 2013-06-13].

[324] M. Wu, R. C. Miller, and S. L. Garfinkel. Do security toolbars actually prevent phishing attacks? In *Conference on Human Factors in Computing Systems*, CHI '06, pages 601–610. ACM, 2006.

[325] M. Wu, R. C. Miller, and G. Little. Web wallet: preventing phishing attacks by revealing user intentions. In *Symposium on Usable Privacy and Security*, SOUPS '06, pages 102–113. ACM, 2006.

[326] T. Wu. The secure remote password protocol. In *Network and Distributed System Security Symposium*, NDSS '98, pages 8:1–8:15. Internet Society, 1998.

[327] G. Xiang, J. Hong, C. P. Rose, and L. Cranor. CANTINA+: a feature-rich machine learning framework for detecting phishing web sites. *Transactions on Information and System Security*, 14 (2), pages 21:1–21:28. ACM, 2011.

[328] G. Xiang and J. I. Hong. A hybrid phish detection approach by identity discovery and keywords retrieval. In *Conference on World Wide Web*, WWW '09, pages 571–580. ACM, 2009.

[329] G. Xiang, B. Pendleton, J. Hong, and C. Rose. A hierarchical adaptive probabilistic approach for zero hour phish detection. In *European Symposium on Research in Computer Security*, ESORICS '10, pages 268–285. Springer, 2011.

[330] E. Ye, Y. Yuan, and S. Smith. Web spoofing revisited: SSL and beyond. Technical report, Dartmouth College, 2002.

[331] K.-P. Yee. Aligning security and usability. *Security & Privacy*, 2, pages 48–55. IEEE, 2004.

[332] K.-P. Yee and K. Sitaker. Passpet: convenient password management and phishing protection. In *Symposium on Usable Privacy and Security*, SOUPS '06, pages 32–43. ACM, 2006.

[333] C. Yue and H. Wang. Anti-phishing in offense and defense. In *Computer Security Applications Conference*, ACSAC '08, pages 345–354. ACM, 2008.

[334] J. Zdziarski, W. Yang, and P. Judge. Approaches to phishing identification using match and probabilistic digital fingerprinting techniques. In *MIT Spam Conference*, pages 1115–1122. MIT, 2006.

[335] Y. Zhang, S. Egelman, L. Cranor, and J. Hong. Phinding phish: Evaluating anti-phishing tools. In *Network and Distributed System Security Symposium*, NDSS '07, pages 5:1–5:16. Internet Society, 2007.

[336] Y. Zhang, J. I. Hong, and L. F. Cranor. Cantina: a content-based approach to detecting phishing web sites. In *Conference on World Wide Web*, WWW '07, pages 639–648. ACM, 2007.

[337] W. Zhu, N. Zeng, and N. Wang. Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS® implementations. In *NorthEast SAS Users Group*, NESUG '10. SAS Institute Inc., 2010.

[338] M. E. Zurko and R. T. Simon. User-centered security. In *Workshop on New Security Paradigms*, NSPW '96, pages 27–33. ACM, 1996.

[339] M. Zusman and A. Sotirov. Sub-prime PKI: attacking extended validation SSL. In *Black Hat USA*. UBM Tech, 2009.

V

Appendix

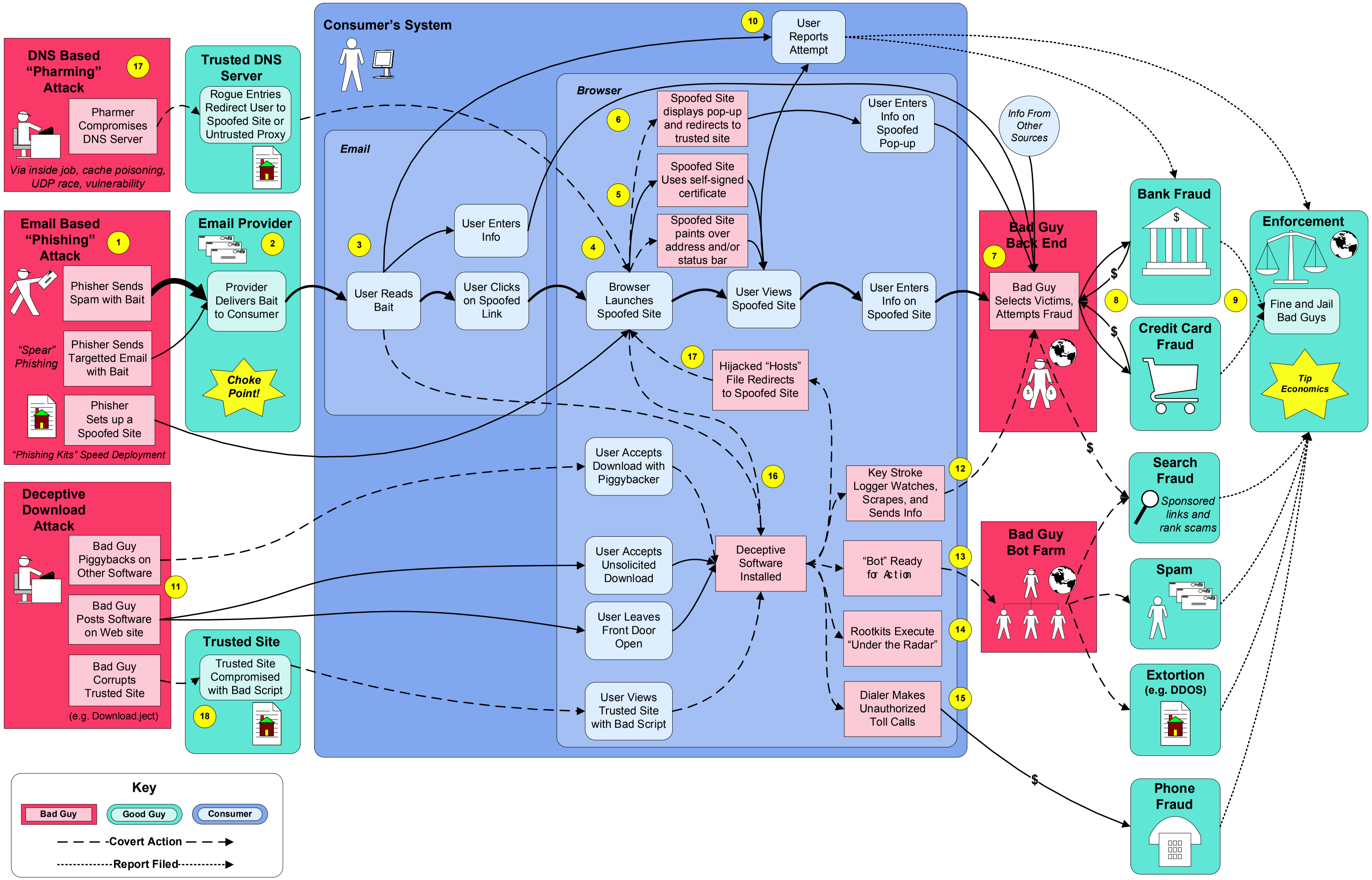
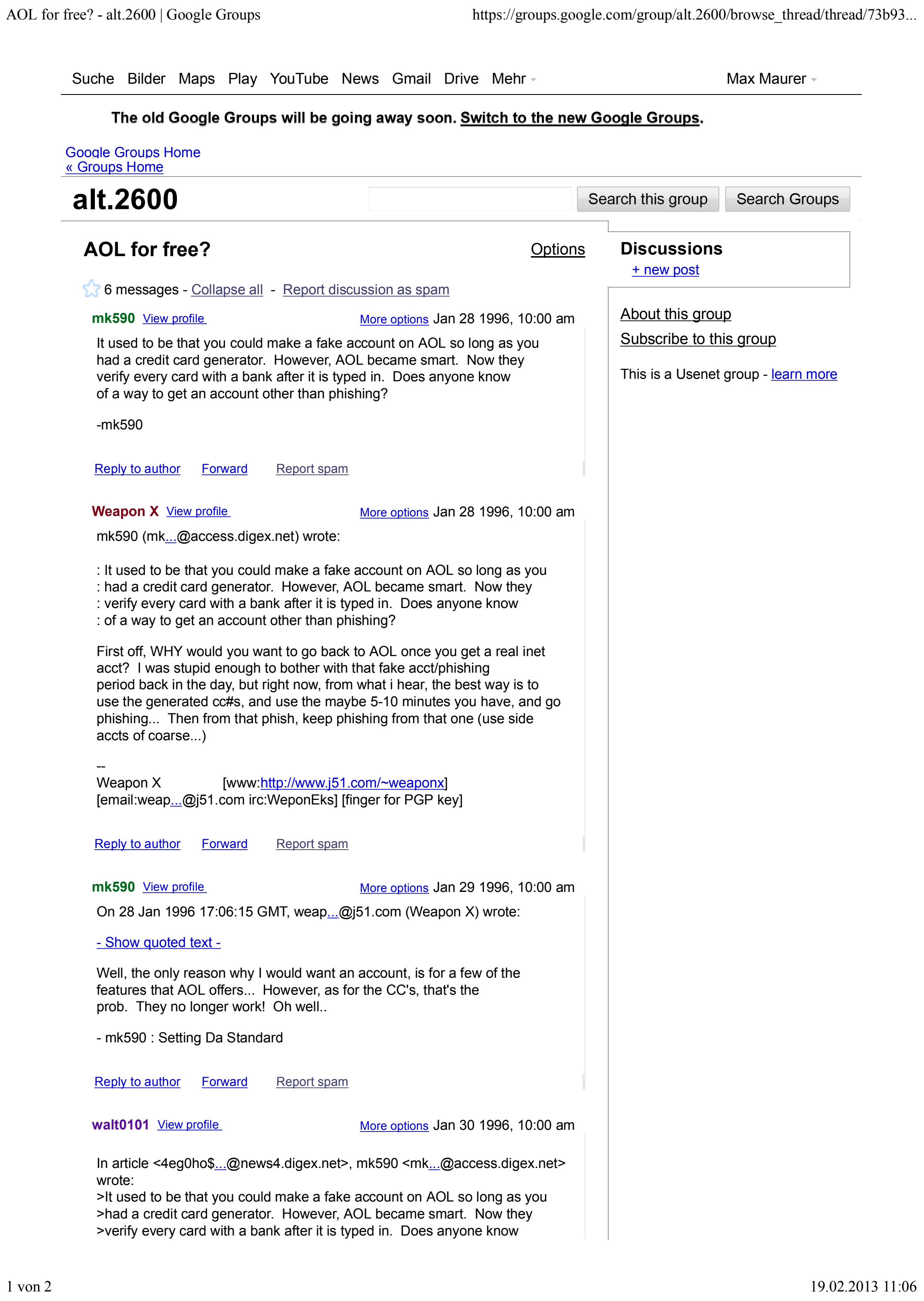

DNS Based “Pharming” Attack
17
Pharmer Compromises DNS Server
Via inside job, cache poisoning, UDP race, vulnerability
Trusted DNS Server
Rogue Entries Redirect User to Spoofed Site or Untrusted Proxy
Email Based “Phishing” Attack
1
Phisher Sends Spam with Bait
“Spear” Phishing
Phisher Sends Targetted Email with Bait
Phisher Sets up a Spoofed Site
“Phishing Kits” Speed Deployment
Email Provider
2
Provider Delivers Bait to Consumer
Choke Point!
Deceptive Download Attack
Bad Guy Piggybacks on Other Software
11
Bad Guy Posts Software on Web site
Bad Guy Corrupts Trusted Site
(e.g. Download.ject)
Trusted Site
Trusted Site Compromised with Bad Script
18
Consumer’s System
Email
3
User Reads Bait
User Enters Info
User Clicks on Spoofed Link
Browser
4
Browser Launches Spoofed Site
5
6
Spoofed Site displays pop-up and redirects to trusted site
Spoofed Site Uses self-signed certificate
Spoofed Site paints over address and/or status bar
User Views Spoofed Site
User Enters Info on Spoofed Pop-up
User Enters Info on Spoofed Site
10
User Reports Attempt
17
Hijacked “Hosts” File Redirects to Spoofed Site
User Accepts Download with Piggybacker
User Accepts Unsolicited Download
User Leaves Front Door Open
User Views Trusted Site with Bad Script
16
Deceptive Software Installed
12
Key Stroke Logger Watches, Scrapes, and Sends Info
13
“Bot” Ready for Action
14
Rootkits Execute “Under the Radar”
15
Dialer Makes Unauthorized Toll Calls
Info From Other Sources
Bad Guy Back End
7
Bad Guy Selects Victims, Attempts Fraud
8
$
$
$
$
Bad Guy Bot Farm
Bank Fraud
Credit Card Fraud
Search Fraud
Sponsored links and rank scams
Spam
Extortion (e.g. DDOS)
Phone Fraud
9
Enforcement
Fine and Jail Bad Guys
Tip Economics
Key
Bad Guy
Good Guy
Consumer
Covert Action
Report Filed

Figure A.1: The “Internet Fraud Battlefield” gives an overview over different attack vectors, consumer vulnerabilities and the possibilities for final fraud [96]



AOL for free? - alt.2600 | Google Groups
https://groups.google.com/group/alt.2600/browse_thread/thread/73b93...
Suche Bilder Maps Play YouTube News Gmail Drive Mehr
Max Maurer
The old Google Groups will be going away soon. Switch to the new Google Groups.
Google Groups Home
« Groups Home
alt.2600
Search this group
Search Groups
Options
Discussions
+ new post
About this group
Subscribe to this group
This is a Usenet group - learn more
1 von 2
19.02.2013 11:06

AOL for free?

6 messages - Collapse all - Report discussion as spam

mk590 View profile — More options Jan 28 1996, 10:00 am

It used to be that you could make a fake account on AOL so long as you
had a credit card generator. However, AOL became smart. Now they
verify every card with a bank after it is typed in. Does anyone know
of a way to get an account other than phishing?

-mk590

Reply to author Forward Report spam

Weapon X View profile — More options Jan 28 1996, 10:00 am

mk590 (mk...@access.digex.net) wrote:

: It used to be that you could make a fake account on AOL so long as you
: had a credit card generator. However, AOL became smart. Now they
: verify every card with a bank after it is typed in. Does anyone know
: of a way to get an account other than phishing?

First off, WHY would you want to go back to AOL once you get a real inet
acct? I was stupid enough to bother with that fake acct/phishing
period back in the day, but right now, from what i hear, the best way is to
use the generated cc#s, and use the maybe 5-10 minutes you have, and go
phishing... Then from that phish, keep phishing from that one (use side
accts of coarse...)

--
Weapon X [www:http://www.j51.com/~weaponx]
[email:weap...@j51.com irc:WeponEks] [finger for PGP key]

Reply to author Forward Report spam

mk590 View profile — More options Jan 29 1996, 10:00 am

On 28 Jan 1996 17:06:15 GMT, weap...@j51.com (Weapon X) wrote:

- Show quoted text -

Well, the only reason why I would want an account, is for a few of the
features that AOL offers... However, as for the CC's, that's the
prob. They no longer work! Oh well..

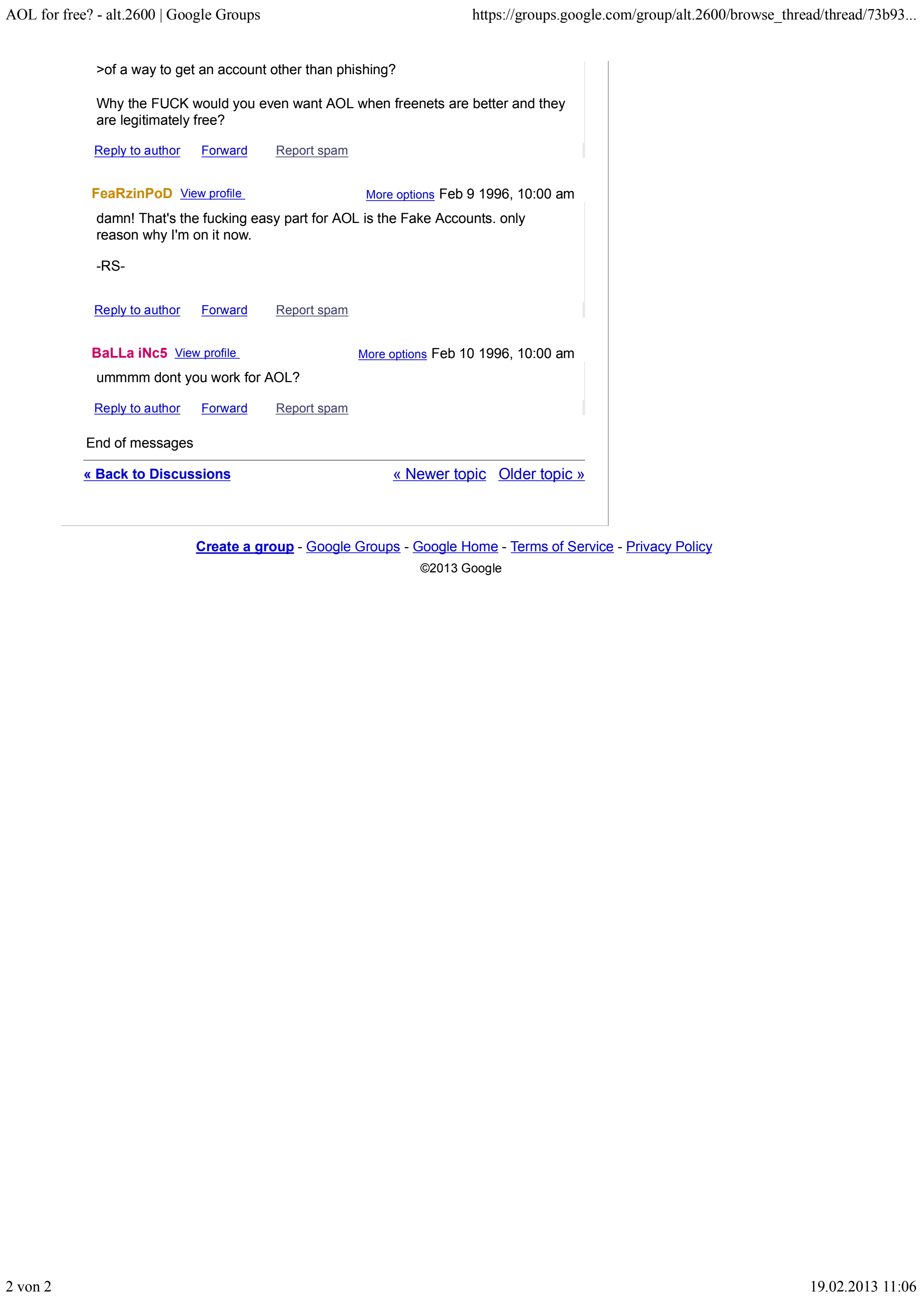
- mk590 : Setting Da Standard

Reply to author Forward Report spam

walt0101 View profile — More options Jan 30 1996, 10:00 am

In article <4eg0ho$...@news4.digex.net>, mk590 <mk...@access.digex.net>
wrote:
>It used to be that you could make a fake account on AOL so long as you
>had a credit card generator. However, AOL became smart. Now they
>verify every card with a bank after it is typed in. Does anyone know

Figure A.2: The full text of the newsgroup post of what is usually referred to as the first official occurrence of the term phishing.



AOL for free? - alt.2600 | Google Groups https://groups.google.com/group/alt.2600/browse_thread/thread/73b93...

>of a way to get an account other than phishing?

Why the FUCK would you even want AOL when freenets are better and they are legitimately free?

Reply to author Forward Report spam

FeaRzinPoD View profile More options Feb 9 1996, 10:00 am

damn! That's the fucking easy part for AOL is the Fake Accounts. only reason why I'm on it now.

-RS-

Reply to author Forward Report spam

BaLLa iNc5 View profile More options Feb 10 1996, 10:00 am

ummmm dont you work for AOL?

Reply to author Forward Report spam

End of messages

« Back to Discussions « Newer topic Older topic »

Create a group - Google Groups - Google Home - Terms of Service - Privacy Policy

©2013 Google

2 von 2 19.02.2013 11:06

Figure A.3: The full text of the newsgroup post of what is usually referred to as the first official occurrence of the term phishing.

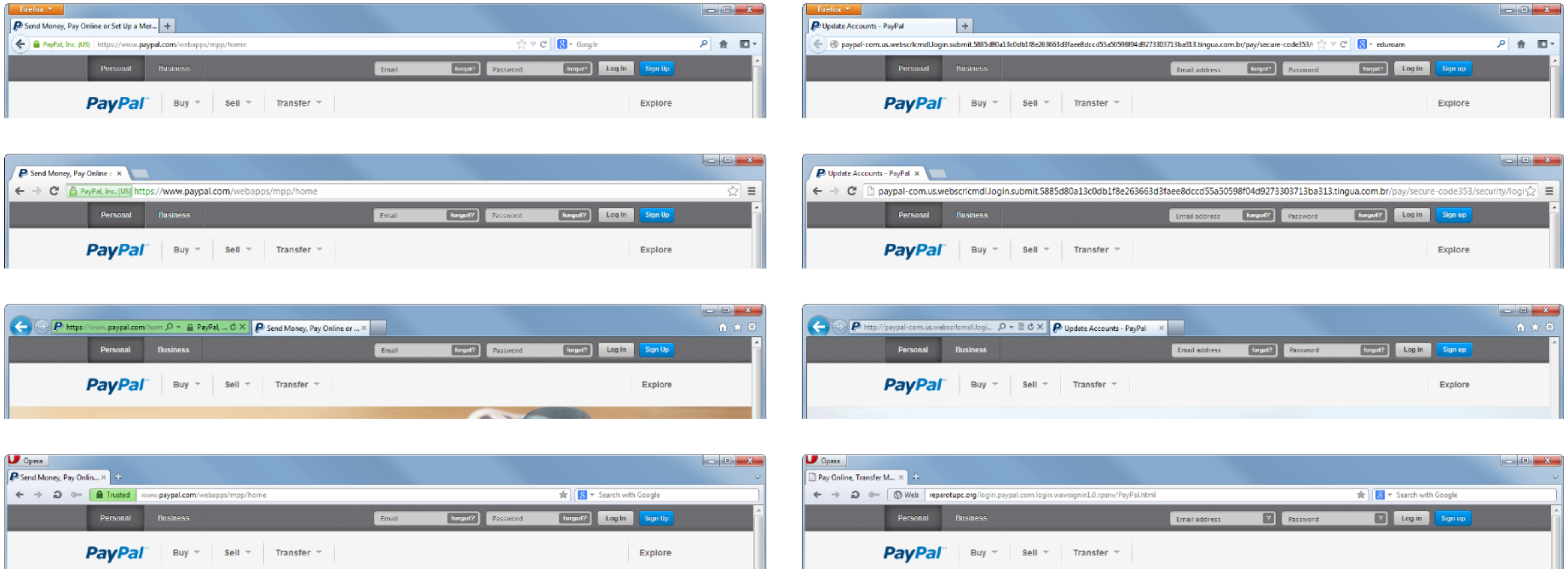

Send Money, Pay Online or Set Up a Mer...
PayPal, Inc. (US) https://www.paypal.com/webapps/mpp/home
Update Accounts - PayPal
paypal-com.us.webscrcmdl.login.submit.5885d80a13c0db1f8e263663d3faee8dccd55a50598f04d9273303713ba313.tingua.com.br/pay/secure-code353/security/login
Trusted www.paypal.com/webapps/mpp/home
Pay Online, Transfer M...
Web reparotupc.org/login.paypal.com.login.wawsignin1.0.rpsnv/PayPal.html
Personal
Business
Email
Email address
Password
forgot?
Log In
Sign Up
PayPal
Buy
Sell
Transfer
Explore

Figure A.4: Large version of the security indicator figure 2.12-

INDEX

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. .5.)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig und ohne unerlaubte Beihilfe angefertigt wurde.

München, den 27. April 2014

Max-Emanuel Maurer