LINFENG LI

A Contingency Framework to Assure the User-Centered Quality and to Support the Design of Anti-Phishing Software

■

ACADEMIC DISSERTATION
To be presented, with the permission of
the board of the School of Information Sciences of the University of Tampere,
for public discussion in the Auditorium Pinni B3107,
Kanslerinrinne 1, Tampere, on March 15th, 2013, at 12 o'clock.

UNIVERSITY OF TAMPERE

UNIVERSITY
OF TAMPERE

ACADEMIC DISSERTATION
University of Tampere
School of Information Sciences
Finland

Distribution
Bookshop TAJU
P.O. Box 617
33014 University of Tampere
Finland

Tel. +358 40 190 9800
taju@uta.fi
www.uta.fi/taju
http://granum.uta.fi

Cover design by
Mikko Reinikka

Acta Universitatis Tamperensis 1810
ISBN 978-951-44-9061-3 (print)
ISSN-L 1455-1616
ISSN 1455-1616

Acta Electronica Universitatis Tamperensis 1286
ISBN 978-951-44-9062-0 (pdf)
ISSN 1456-954X
http://acta.uta.fi

Tampereen Yliopistopaino Oy – Juvenes Print
Tampere 2013

Acknowledgements

This dissertation work took me lots of efforts and time, and it could not be accomplished without supports from my supervisors, colleagues and family. I take this opportunity to express how grateful I feel.

I have been very thankful to having two excellent supervisors, Eleni Berki and Marko Helenius. You always gave me sufficient options when I met challenges on my research or problems in my life. Your encouragement and guidance made me confident of my doctoral research. As a foreign doctoral student of yours, my life and experiences in Finland are very special and impressive. Moreover, the financial supports from Tampere Doctoral Programme in Information Science and Engineering (TISE), Internet of Things project (Tekes Finalnd) and University of Tampere helped me concentrate on my doctoral research.

I cooperated very well with my colleagues, and many thanks are given to Saila Ovaska and Reijo Savola who contributed a lot to my research as usability and security experts. It is also very grateful to work with Timo Nummenmaa, and the research tools you suggested helped my research very much. In addition, it is appreciated to receive many valuable suggestions and comments from Elli Georgiadou and Margaret Ross.

Finally, my special and deepest thanks are given to my family. Every time when I had difficulties or tried to quit, my family encouraged me at all times so that I could persevere with my research. I got married during my doctoral study. My wife, Nasa, did lots of work for this home. You improved my life and made our life more comfortable and colorful.

Dedicated to my family

Linfeng Li

Glossary

Anti-phishing	Methods or solutions to prevent phishing attacks.
API	Application Programming Interface: a specification that is used to interact among different software components
Authentication	An operation used by information systems to confirm the identities of users who have been granted access to their data and systems resources. Usually authentication of a user is based on what the user knows, is or has.
Backdoor	A type of online attack that circumvents the system protections and provides access to infected computers for attackers to visit data and use resources on the infected computers. After targeted computers are compromised, backdoor remains undetected. After backdoor is deployed, the intruders can then silently monitor victims' activities or launch various attacks, e.g. destroying or altering files, stealing information etc.
Contingency approach	A research method, in which scholars are able to investigate a research domain from different perspectives, due to the fact that a single research method is not adequate to study the uncertainty and unpredictable factors in the research domain.
Cross-site scripting	A type of online attack that injects malicious web page scripts or other codes into vulnerable web pages to collect victims' personal information. These vulnerable web pages are usually trusted by victims.
Delegation Signer	In DNSSEC infrastructure, a type of domains whose public key is stored in the caches of its parent DNS servers using
DNS poisoning	A type of digital attack that aims to reroute a DNS request for a web page, cause the name server to return an incorrect IP

	address, and divert traffic to another computer.
DNS resolver	The client-side of the DNS which is to initiate and sequence the DNS queries.
DNSSEC	A security infrastructure for DNS servers
DNS server	Domain Name System server: a server that offers information to map domain names and IP addresses.
DOM	Document Object Model: a model of representing and interacting with objects on web page documents.
Electronic token	An electronic compact device in which secret or credential information is stored. Users are able to use this device for authentication.
Exploit	An exploit is a piece of software, a chunk of data, or sequence of commands that makes use of a weakness or vulnerability at the target system in order to cause unintended behavior or damage to occur at the target system.
Feature	A feature can be a word or a phrase in a text. The features in a text can be collected to classify the category of the text.
Feature size	The number of features selected to be learned or examined by content algorithms.
Hacker	Someone who breaks into a computer system by circumventing its security protections.
Hash	A data transformation that is used to convert a variable-size data input into a fixed-size string. The output string is called hash value. In DNSSEC, it is a hash function, e.g. SHA-1.
Heuristic	In the usability research field a heuristic is a list of pre-defined detailed usability metrics and severity levels of usability problems for assisting usability experts in their evaluations.
Hook	A technique to read the system messages from the message queue of Microsoft systems.
ICANN	Internet Corporation for Assigned Names and Numbers, an organization to coordinate the Internet resources, e.g. IP

	address
IP	Internet Protocol, a protocol designed for the use in interconnected systems of packet-switched computer communication networks. This protocol defines the IP datagram structure and how to route the datagram.
Key Signing Key	In DNSSEC infrastructure, an authentication key that corresponds to a private key to sign one or more other authentication keys for a given zone.
MAC	Message Authentication Code: in the cryptography, it is secret information to authenticate a message and to provide integrity and authenticity assurances on the message.
Malware	Malicious program is intentionally developed to harm the targeted computer system or steal victims' personal information.
Man-in-the-Middle attacks	A type of attacks that an attacker intercepts the connection and selectively modifies communicated data to masquerade as one or more of the entities involved in a connection.
Misuse cases	Use cases that attackers perform to abuse targeted information systems.
Naive Bayesian	A simple probabilistic classifier based on Bayes' theorem with independence assumptions. Naive is called because of the strong independence assumptions.
Pharming	A type of online threat that redirects victims' DNS requests to malicious and fraudulent websites so as to mislead victims to provide their online credentials on these websites.
Phishers	Criminals who design and conduct phishing attacks.
Phishing	A type of online attack to use information technology and alluring information to mislead victims to provide their personal information to its attackers.
Phishing-resistant system	An information system that is able to prevent users from being spoofed by phishing scams.

Phishing scam	A set of tricks designed by phishers to successfully conduct phishing attacks.
PIN	Personal Identification Number.
PKI	Public-Key Infrastructure: an asymmetric-encryption-based infrastructure to create, manage, distribute, use, store and revoke public key certificates.
Risk	A risk is an expectation of loss when a particular threat causes harmful results of targeted computers or information systems.
Rootkit	A stealthy type of software, often malicious, designed to hide the existence of certain processes or programs from normal methods of detection and enable continued privileged access to a computer.
Short Message Service	A text messaging service component of cell phones.
Social engineering	A type of attack that takes advantage of cognitive bias in the human decision-making process so as to convince victims to believe in the information presented by the attackers.
Spam	Unsolicited messages for advertisement or scams.
Spyware	A type of malware that sends the information about the contents, status, or operation of the computer to a remote system or a user.
SSL	Secure Sockets Layer Internet protocol: originally developed by Netscape that uses connection-oriented end-to-end encryption to guarantee the data confidentiality and data integrity on the Internet.
SQL	Structured Query Language.
SQL injection	A type of online attack that exploits the vulnerabilities in the web application program code accessing the database of an online information system. It allows an entire SQL query or portions of it as a parameter, issued through a field or fields of a form.

TCP	Transmission Control Protocol: a transport layer protocol on the Internet for using a handshake mechanism to guarantee a connection.
Threat	A threat is a circumstance or an event which may potentially harm a system and cause some security damage, including destruction, data loss, financial loss etc.
TLS	Transport Layer Security is an Internet protocol between transport layer and application layer, and uses connection-oriented end-to-end encryption to guarantee the data confidentiality and data integrity on the Internet. SSL is the predecessor of TLS. In TLS, the security has been improved.
URL	Uniform Resource Locator: a means of identifying a resource and locating the resource on the Internet.
User Account Control	A security infrastructure built in Microsoft's Vista and later systems to manage the system administrative privileges.
Virus	Program code that has a capability to replicate by itself (Helenius 2002, p.12). A virus may contain one or more payloads. A payload is a malicious operation contained in the virus that is triggered under certain condition.
Vulnerability	A vulnerability is a weakness which an attacker uses to attack targeted software, computers or information systems.
Worms	A worm is a virus that replicates itself from computer to computer through a network. A worm may or may not contain a payload. A payload is a malicious operation contained in the virus that is triggered under certain condition.
Zone Signing Key	In DNSSEC infrastructure, an authentication key that corresponds to a private key to sign a zone

Abstract

Anti-phishing software is widely used in online security and privacy. At the same time, phishing attacks are becoming increasingly advanced. From a software-engineering point of view, the growing sophistication of phishing attacks means that anti-phishing software must be continually improved and constantly updated. Software quality engineers assert that quality improvement is more effective at the early lifecycle stages such as during requirements elicitation. Hence, in this study, the investigation of end users' preferences and behaviors can help to deduce usability and security requirements that are essential to the design quality of anti-phishing software and phishing-resistant systems. In so doing, this research studies and analyzes current phishing scams, examines online service user interfaces that are vulnerable to phishing attacks, and attempts to document and model human online behavior.

Online user behavior, like human nature, is varied and unpredictable because of unforeseen events and various other factors. In phishing attacks, these events and factors are exploited. Thus, to provide good quality anti-phishing software and phishing-resistant systems, exploited online user behavior needs to be researched. Software quality engineers use contingency methodologies to analyze and model unanticipated events and factors that are vital for designers of anti-phishing technologies. Therefore, this thesis introduces a research framework consisting of user-centered quality assurance theories, engineering methodologies, heuristic evaluation, and usability experiments. In addition to a contingency approach to be utilized as a guide in situations of high uncertainty, concluding remarks are made regarding industrial field research findings on how to improve the design of anti-phishing software and phishing-resistant systems. Thus, as key novel contributions, the thesis offers (i) a set of usability metrics for authentication mechanisms and (ii) an improved understanding of online social behavior. Instead of establishing sociological or psychological theory of online social behaviors, this knowledge may significantly assist in the design of high-quality anti-phishing software for humans online. Thus software development and deployment is driven with preventive rather than reactive anti-phishing strategies in mind.

Keywords:

Security, privacy, software quality, phishing, identity theft, usability, authentication, metrics, human behavior, contingency approach/strategy, heuristic evaluation, requirement elicitation, misuse case, content-filtering performance, end-user survey.

Table of Contents

1. Introduction

The Internet enriches peoples' lives with its variety of web-based services and communication tools, while end users are required to provide personal information to obtain better online services (Dhillon and Moores 2001). For example, we purchase a variety of products from online retailers; we talk and participate in video chats with our friends and families; and we play games online together with other players globally. Such facilities bring convenience and pleasure to our daily lives.

However, web applications associated with users' personal information are also misused by individuals, corporations, government agencies and organized criminals (Garfinkel 1995, Warren and Hutchinson 2002, p.126, Semenov et al. 2011a, p.232). Such attackers are not only interested in information society infrastructures but also in personal private information and financial benefits. These attacks have been analyzed and debated from the perspective of human morality and information ethics. For example, Duquenoy and her colleagues (1999) discuss them with Haberma's theory of Discourse Ethics. They believe that these attacks are the main ethical issue of the Internet, and those cyberspace ethics are different from conventional ethics. They also advocate that cyberspace ethics should be regulated internationally. Siponen (2004, p.287) also addresses information ethics and concludes that the makers of malware (one type of artificial agent) should be blamed. He further argues that it is not adequate to use human morality to ensure security and privacy (Siponen 2003, p.243). It is, therefore, clear that users need protection against such online attacks.

Although security researchers have made great efforts in the fight against attacks such as SQL injections, cross-site scripting, worms and backdoor techniques (Askola et al. 2008, Eronen et al. 2009, Christey 2011, F-secure 2012, Scarfone et al. 2012, Syroid 2012), online crimes have not ended. One of these types of online crimes is phishing. So far, many researchers have given their definitions on phishing. Dhamija and her colleagues (2006) define phishing as the practice of directing users to fraudulent web sites. This definition only emphasizes the consequences of phishing attacks, but the technologies used for phishing are not presented. While modeling phishing attacks, Jakobsson (2006) describes that phishing is the marriage of technology and social engineering. However, this description does not mention the consequences of phishing attacks. A more detailed definition (Zhang et al. 2007) is next presented:

"Phishing is a type of semantic attack in which victims are sent emails that deceive them into providing account numbers, passwords, or other personal information to an attacker. Typical phishing emails falsely claim to be from a reputable business where victims might have an account. Victims are directed to a spoofed web site where they enter information such as credit card numbers or Social Security Numbers."

This is a more detailed definition. According to this, phishing is classified as a semantic attack. The purpose of this semantic attack is to collect victims' credential information, and also, in this definition, emails are the only medium to deliver the convincing and fraudulent information to the victims. However, emails are not the only way to send phishing information, but also other media are being used, e.g. a video clip (Tencent Service 2012), social media web sites (Chhabra et al. 2011, Hong 2012). In addition, phishing attacks can also be carried out by the attackers who are able to install the malware to monitor and collect victims' passwords and usernames (APWG 2012). Based on my observations and research on phishing attacks, I define phishing as follows:

> *"Phishing is a type of online attack to use information technology and alluring information to mislead victims to provide their personal information to its attackers."*

In this definition, it is emphasized that the purpose of phishing is to collect victims' personal information, including usernames, passwords, personal email addresses, personal identity numbers, or mobile phone numbers, and the list can go on. Furthermore, according to this definition, a phishing attempt uses information technology and alluring information to attack. Some information technology can be employed by attackers to deliver phishing information, e.g. emails, social web sites, or instant messengers. Some other advanced information technology can be used by, e.g. key loggers, which can record a victim's keystrokes on the keyboard. Alluring information sent by phishers is to build the social relationship and the trust relationship between phishing attackers and victims.

Phishing seriously challenges and collapses trust in electronic commerce and e-service security (Litan 2004, Dinev 2006). In this sense, phishing is a severe threat to e-commerce services and web application providers. According to the APWG (2012), the number of phishing attack reports reached 53,939 in March 2012. The loss to the economy reached $687 million in the first quarter of 2012 (RSA 2012).

Phishing prevention protects users against such attacks, through anti-phishing software. In this thesis, anti-phishing software refers to a stand-alone application to prevent phishing attacks, and it is also referred to as a phishing-resistant information system or a phishing-resistant feature of an information system whose users phishing attackers cannot deceive. For example, anti-phishing toolbars installed at web browsers are one type of anti-phishing software, and a feature to verify users to log in to an information system can also be called anti-phishing software. Different anti-phishing software has been developed, but the quality of the software is not satisfactory (e.g. Asokan et al. 2003, Wu et al. 2006b, Zhang et al. 2007, Wang et al. 2008, Shahriar and Zulkernine 2010). Therefore, it is needed to study how to design high-quality anti-phishing software.

Various models and theories have been proposed to facilitate the design of secure information system (e.g. Hutchinson and Warren 2000, Siponen et al. 2006, Kajava et al. 2006, Siponen and Heikkar 2008, Veijalainen 2007, Veijalainen and Hara 2011), but few studies have investigated how to improve the design of secure information systems from the perspective of end users. I believe that, without enough attention on end users, it is hard to design high-quality anti-phishing software. There are two reasons for this. Firstly, in the information society environment, different users may understand one type of technology differently. This difference may bring different security and privacy risks such as abuse of the technologies (Duquenoy 2007). This implies that anti-phishing, as one type of technology in the information society environment, should be studied from the perspective of end users. Secondly, it is imperative to put attention to users' requirements when anti-phishing software is designed. Taking phishing into consideration, the security and the privacy (as basic quality features) of information systems need to be addressed and more systems stakeholders should be involved (Nikander and Karvonen 2001). Therefore, a requirements analysis of these stakeholder groups is needed. These stakeholder groups should include end users of anti-phishing software, other people who are social connections of these end users, the designers of anti-phishing software, the phishing scams, the existing vulnerabilities of the current authentication methods and other information.

To understand these different stakeholder groups in the phishing context, the research approach should be also carefully selected. As Jakobsson and Ratkiewicz (2006, p.513) stated, *"Phishing is a multi-faceted techno-social problem, and there is no known silver bullet."* This implies that, in order to design high-quality anti-phishing software, it is also needed to investigate phishing and anti-phishing research domains from multiple perspectives. In this way, the different user-centered quality views and requirements from different stakeholders can be discovered. Since it is not possible to use the same research method to study different stakeholders from different perspectives, I believe that a contingency approach is more suitable than other research approaches when studying aspects of the user-centered quality assurance and the design requirements of anti-phishing software.

Therefore, the research scope of this thesis research covers two objectives, i) to investigate phishing from multiple points of view and ii) to introduce a contingency research framework to assure the user-centered quality and to support the design of anti-phishing software.

Contingency theory was established by Fiedler (1963) to help management in an uncertain and dynamic context. This theory was borrowed and expanded in the studies of information system research and development. For example, a contingency approach was used to discover the relationships between contextual factors and decision process attributes leading to the strategic applications of information systems (Sabherwal and King 1992). Saleem (1996) used a contingency approach to compare different types of users and concluded that user expertise is a useful criterion for selecting participants to serve on design teams and for determining the appropriate extent of a participating user's influence on system design. With a contingency based approach, Land (1998) classified different dominant systems and suggested that different types of systems require different processes of requirements elicitation, design, development and implementation. Similarly, Lin and his co-workers (2000) applied a simultaneous contingency approach to confirm the positive link between user participation and system success.

Wang and his colleagues (2012) employed a contingency method to compare online decision aids with different user-system interaction modes.

Contingency approaches help information systems designers to solve development problems and deal with dynamics and uncertainty (e.g., Naumann et al. 1980, Avison 1990, Tuunanen et al. 2007). One representative contingency approach for information systems is Multiview methodology (Episkopou and Wood-Harper 1985, Avison 1990, p.60). Instead of only addressing technologies, the Multiview methodology, using different research methods and tools, investigates social, human and organizational issues of information systems development.

A contingency approach is suitable for the research described in the present thesis for a number of reasons. Firstly, the research of this thesis studies anti-phishing from a design perspective. Secondly, phishing (and anti-phishing) is a dynamic environment that is full of unpredictable events and factors. For example, phishers may use different information technologies to circumvent security protection. Anti-phishing researchers can hardly predict how next information technologies will be abused by phishers. Users with different experiences of using Internet may have their own understanding on how to protect online personal information. It is also uncertain how anti-phishing software can be designed to protect different users. In this thesis, the contingency approach taken to the problem of phishing and the consequent anti-phishing software design is supported by user-centered quality assurance theories, engineering methodologies, heuristic evaluation and usability experiments. This research approach, that is considering solutions depending on particular situations, is novel in the context of anti-phishing design and has not been adopted as an analysis and design strategy proposed in the anti-phishing research area earlier.

About the thesis author's previous research

The research topic of the thesis is divided into several subordinate research questions. These research questions can be categorized into two groups based on the methodologies used to improve the quality of anti-phishing software and phishing-resistant systems. In this thesis, improvement was either based on the software-engineering methodologies (research questions 1 and 2 in Table 1) or based on end-user studies (research questions 3, 4, 5, 6 and 7 in Table 1). The various research methods used and the answers found are listed in Table 1. Different research methods were used for different research questions. This also shows that a contingency approach is a suitable and feasible research framework for this thesis.

Table 1. The research questions, methods and answers

Research 1:	**Question:**	How can an information system be designed, validated and documented against phishing with the misuse case method?
	Method:	Theory-testing case-research (Järvinen 2012, p.58).
	Answers:	The design quality features of the misuse case method can offer reliability to adequately prevent system users from phishing. However, this method requires designers with sufficient experience in system design and system security.
Research 2:	**Question:**	How can a reliable performance evaluation on spam/phishing content filtering be designed?
	Method:	Literature review (Järvinen 2010).
	Answers:	Feature selection methodology, feature size and misclassification cost factor are the key variables in the design of content-filtering experiments. Language issues, skewed corpora topics, and intelligent adversaries are the barriers when designing reliable performance evaluation on spam/phishing content filtering.
Research 3:	**Question:**	How severe and annoying are malicious messages, and why?
	Method:	Field methods, end-user survey (Järvinen 2012, p.54).
	Answers:	The survey showed that sophisticated spam/phishing messages can cause time loss, disturbance and danger. Further research is needed on software user psychology, human-centered software design quality criteria and the cognitive profiles of software/email exploiter.
Research 4:	**Question:**	What are the general usability design principles for

		anti-phishing client-side applications?
	Method:	Evaluation of construction results (Järvinen 2012, p.115).
	Answers:	Important usability issues were found from the tested applications. A design method was determined involving three key components (warnings, help system and main user interface) on the anti-phishing toolbar.
Research 5:	**Question:**	Can a blacklist-based anti-phishing toolbar help users to identify more phishing pages than a whitelist-based one?
	Methods:	Evaluation of construction results (Järvinen 2012, p.115), action research (Järvinen 2012, p.125).
	Answers:	User behavior in a phishing context was observed. There was no significant difference between blacklist-based and whitelist-based anti-phishing toolbars in the extent to which they helped users to identify some phishing pages. Users can identify more phishing pages after sufficient education on phishing and anti-phishing toolbars.
Research 6:	**Question:**	What do end users expect from anti-phishing software and phishing-resistant systems?
	Method:	Literature review (Järvinen 2010).
	Answers:	Previous research on anti-phishing was reviewed. The user requirements for the anti-phishing software and phishing-resistant systems were collected. The methodologies in the previous research were addressed.
Research 7:	**Question:**	What are the usability metrics for designers of online authentication mechanisms?
	Method:	Design research, building process (Järvinen 2012, p.101).
	Answers:	Nine usability metrics were deduced based on the

		analysis of existing authentication mechanisms and anti-phishing research.

Introduction to the thesis publications

The author of the present thesis, who was also first author of the below-listed publications, was the main researcher in charge of designing and conducting all the research activities. The corresponding peer-reviewed publications are listed based on the order of the research questions given in Table 1:

1. Li L., Helenius M., Berki E. (2007). Phishing-Resistant Information Systems: Security Handling with Misuse Cases Design, *Proceedings of Berki, E., Nummenmaa, J., Sunley, I., Ross, M. & Staples, G. (Eds) Software Quality in the Knowledge Society*, SQM 2007. Tampere, Finland,1-2 August 2007, pp. 389-404.
2. Li L., Berki E., Helenius M. (2011). Evaluating the Design and the Reliability of Spam/Phishing Content Filtering Performance Experiments, *Proceedings of Dawson, R., Ross, M., Staples, G. (Eds), Global Quality Issues, SQM 2011*, Leicestershire UK, 18 April 2011, pp.339–357.
3. Li L., Helenius M., Berki E. (2011). How and Why Phishing and Spam Messages Disturb Us? *Proceedings of Bradley G. (Ed) IADIS International Conference ICT, Society and Human Beings 2011*, Rome, 20-26 July, 2011, pp.239–244.
4. Li L., Helenius M. (2007). Usability Evaluation of Anti-phishing Toolbars, *Journal of Computer Virology* (3), pp.163–184.
5. Li L., Berki E., Helenius M., Ovaska S., (2012). Towards a contingency approach with whitelist- and blacklist-based anti-phishing applications: what do usability tests indicate? Submitted to: Behaviour & Information Technology Journal, (accepted, to be published).
6. Li L. (2012). Overview of User-centered Quality Assurance Methodologies for Anti-phishing Software and Phishing-resistant Systems, *Proceedings of Berki, E., Valtanen, J., Nykänen P., Ross, M., Staples, G., Systä K. (Eds), Quality Matters*, SQM 2012, Tampere, Finland, 20-23 August 2012, pp. 11-20.
7. Li L., Berki E., Helenius M., Savola R. (2012). New Usability Metrics for Authentication Mechanisms, *Proceedings of Berki, E., Valtanen, J., Nykänen*

P., Ross, M., Staples, G., Systä K. (Eds), Quality Matters, SQM 2012,, Tampere, Finland, 20-23 August 2012, pp. 239-250.

To investigate phishing and its dynamic evolution research domain with a contingency approach, a research framework is required to look into the entities in the phishing context. This research framework, as briefly explained earlier, consists of various research methods: user-centered quality assurance theories (Papers 3 and 7 in the above list), engineering methodologies (Papers 1, 2 and 6), heuristic evaluation (Paper 4) and usability experiments (Paper 5). Papers 1 and 7 concentrate on phishing-resistant online services. In addition to the research framework, some industrial field research findings are presented in these papers. The key novel contributions are (i) a new set of usability metrics for online authentication mechanisms (proposed in Paper 7) and (ii) a conclusion on the improved understanding of the online user social behavior (in Paper 6). The contributions of the papers are briefly introduced in the following sections.

We firstly designed a set of misuse cases (requirements) and tried it in the context of a system of online phishing. Based on our observations in this research, we emphasized the importance of user-centered quality assurance for phishing-resistant systems, although misuse cases can help to elicit some anti-phishing security requirements.

Further, an end-user survey was conducted to investigate how phishing/spam messages disturb people’s daily lives. Upon analysis of the investigation results and survey feedback, it can be concluded that a drastic and influential approach towards the protection of email users needs to emerge by considering combining three important issues: (i) software user psychology; (ii) human-centered software design quality criteria and (iii) the cognitive profiles of software/email exploiters. In this research, the first author analyzed the reported samples, and we jointly designed the survey.

Another method employed in the papers was a literature review. This summarized methods of designing a reliable performance experiment for content-filtering algorithms. Many performance experiments for adaptive algorithms have been conducted, but this was the first literature review to discuss the design of performance experiments. Based on this research, we determined the strengths, weaknesses and limitations of many performance evaluation experiments of email content-filtering methods and how to design a reliable performance experiment for content-filtering algorithms.

With two usability research methods, a heuristic evaluation and a usability test, we also inspected the usability problems of several selected anti-phishing toolbars for web browsers. From these usability research experiments, a large amount of hidden usability issues and security requirements were collected and reported, which could provide valuable suggestions for the future design of anti-phishing software. The first author's main tasks in these two usability evaluations included designing the usability research, conducting the usability test and analyzing the usability evaluation results.

Because of the evolvement of phishing attacks, the quality, usability in particular, metrics for phishing prevention also need to be improved. Therefore, we conducted another key joint research task which aimed at establishing a new set of usability metrics for authentication mechanisms. In this research, nine new usability metrics were described that can both facilitate the future usability evaluation of new authentication mechanisms and act as a guide for designers of the next generation of authentication mechanisms.

Based on these research findings, I was able to provide an overview of user-centered quality assurance methodologies for anti-phishing and phishing-resistant systems. This overview gathered and analyzed each methodology of our previous user-centered quality assurance studies as a valuable reference for future relevant research, particularly for security- and usability-critical software.

The thesis author's contributions in the thesis publications

I wrote most of the text of the following papers, and I was the main author. The co-authors had an advisory role providing guidance, relevant references, supervising and participating in the research process.

In Paper 1, I discovered and described the different types of phishing attacks, and I classified and introduced their prevention methods. In addition, I together with other co-authors considered and demonstrated the misuse cases in the design of an online music purchase information system.

In Paper 2, I gathered and collected representative information on the existing performance experiments and the publications on different text-content-filtering algorithms. I also highlighted the differences and the limitations of the design of the existing performance experiments.

In Paper 3, I together with my co-authors designed the experiment. After the participants' feedback was received, all authors together analyzed the feedback and concluded the findings.

In Papers 4 and 5, I designed and developed the test subject, Anti-phishing IEPlug. I also helped to invite many usability experts and ordinary end users to participate in the heuristic evaluation and usability test. Besides that, I designed most of the heuristic check list and usability test tasks, and observed and analyzed the participants' behaviors and feedback. Together with other co-authors, we collected valuable findings and composed the article.

In Paper 6, I, as the only author of the article, reviewed all the used research methods on the user-centered quality assurance methodologies for anti-phishing software and phishing-resistant systems. I also criticized each of these methodologies and drew conclusions.

In Paper 7, I collected representative information of existing authentication mechanisms, and analyzed these mechanisms based on their characteristics. The analysis work was completed with the collaboration of other authors. Furthermore, I was the main contributor of the design of the new usability metrics for authentication mechanisms in the paper.

The structure of the thesis

The remaining chapters are presented as follows. Chapter 2 introduces phishing, its prevention and online service user interfaces. Chapter 3 describes software quality assurance metrics and methodologies. The author's contributions are briefly explained in Chapter 4. Finally, Chapter 5 concludes with suggestions for quality assurances on anti-phishing applications and phishing-resistant systems.

2. Phishing and Online Services

In designing anti-phishing software and phishing-resistant systems, it is imperative to carefully examine phishing scams and online services exploited by phishers. In this chapter, typical types of phishing scam are introduced based on our observations of phishing samples reported in the media and participants who were invited to our survey. I then describe the user interfaces and authentication mechanisms of online service systems that are frequently involved in phishing scams. Examples of phishing prevention systems are then presented.

2.1 Phishing Scams

In the following, typical phishing scams are described according to their popularity and complexity. New phishing scams are introduced chronologically.

Plain-text phishing emails with no hyperlinks or personal information requested

This type of phishing resembles regular emails. Instead of asking for personal information, phishers usually present themselves as poor or unfortunate people requiring help. For example, a phishing email may resemble a real inquiry for solving life difficulties, but the phisher is able to confirm actively harvested email addresses, build a relationship with the email recipient and take advantage of the relationship to ask for more personal information. One typical example is in Table 2:

Table 2. Plain-text phishing email example with no hyperlinks or personal information requested

```
Return-Path: <elenaga@eposta.ru>
Received: from mailanen.uta.fi ([unix socket])
          by uta.fi (Cyrus v2.3.8) with LMTPA;
          Mon, 07 Jan 2008 20:08:41 +0200
X-Sieve: CMU Sieve 2.3
```

Received: from cd440732a.cable.wanadoo.nl (cd440732a.cable.wanadoo.nl [212.64.115.42])
by mwinf6208.orange.nl (SMTP Server) with SMTP id 0BB071C00084;
Mon, 7 Jan 2008 18:47:22 +0100 (CET)

X-ME-UUID: 20080107174723479.0BB071C00084@mwinf6208.orange.nl

From: "Elena" <elenaga@eposta.ru>

To: <elenaga@eposta.ru>

Subject: Hi

Date: Mon, 07 Jan 2008 20:47:17 +0300

Message-Id: <20080107174722.0BB071C00084@mwinf6208.orange.nl>

X-Greylist: Sender DNS name whitelisted, not delayed by milter-greylist-3.0 (apila.uta.fi [153.1.1.41]); Mon, 07 Jan 2008 20:08:40 +0200 (EET)

MIME-Version: 1.0

Content-Type: TEXT/PLAIN; charset=ISO-8859-1

Content-Transfer-Encoding: 8BIT

X-Spam-Status: LOW ; 39

X-Spam-Level: ***++++++++++

X-Spam-Report: AWL,BAYES_99

X-Scanned-By: MIMEDefang 2.62 on 153.1.1.42

Hi,

My name is Elena, I have 31 years old and I live in Russian province.

I have a 6-years daughter, her father abandoned us and we live with my mother. Recently my mother lost job due to old age and our situation became very difficult.

During the last months the prices for gas and electricity became very high in our region and we cannot use it to heat our home anymore.

The weather is minus 19 degree Celsius already and it become colder each day. We very afraid and we don't know what to do.

The only accessible way for us heat our home is to use portable oven which give heat with burning wood. We have a lot wood in our region and this oven will heat our home all winter for minimal charges.

I work in library and after my job I allowed to use computer. I finded your address in internet and may be you can help us.

We need portable wood burning oven, but we cannot buy it in our local market because it cost equivalent of 194 Euros and is very expensive for us. May be you have any portable oven which you don't use anymore, we will be very grateful to you if you can donate its to us and organize transport of its to our address. This ovens are different, usually they made from cast-iron and weight between 100-150kg.

I hope that you will write me back. I wish that the New Year will bring you happiness and good health.

Elena.

Russia.

In this example the actual domain name of the sender's email address displayed in the field "Message-Id" is not the same as that in the field "From". This difference clearly shows that this is a phishing attempt. In any case, the purpose of this type of phishing email is to validate harvested email addresses; phishers also attempt to collect as much information about the email recipient as possible, such as personal characteristics and interests to build a relationship.

Phishing emails with personal information requested or with hyperlinks to phishing websites where no malware is hosted

This type of phishing pretends to be from acquaintances or services that the victims may use. Similar to the previous type, this type of phishing email attempts to mislead potential victims into certain urgent fabricated situations such as problems with a server or with their personal lives. To solve these problems, victims are asked to provide financial help or personal information. To convince them to provide personal information, phishing emails use the same or similar visual effects of authentic web services, and the contents of such emails are closely related to recent well-known events or incidents. The role of the phishing web site is to provide the same or similar visual effects of authentic web sites to deceive the victims and to collect the private information of the victims.

Phishing emails with malware attachments or redirections to websites where malware is hosted

This type of phishing scam usually does not directly ask for financial assistance or personal information. Instead, these phishing emails interest victims either because of that they are sent from supposed acquaintances or because of their interesting contents. Some content may be multimedia data recording a memorable time, or some may contain no text other than a link from supposed acquaintances that require the victims to click on it to find out more. No matter what contents are presented, the two most basic goals are to intrigue the victims and to install malware on their computers. The purpose of the malware at the phishing web sites is to stealthily monitor and collect the private information of the potential victims.

Phishing scams through communication tools other than emails

Originally phishing scams were spread via emails. Now there are more choices and greater convenience for online communications and business, e.g., social networking and online instant messaging. To take advantage of this, phishers designed scams on these new online services and platforms. Usually, the goals of this type of phishing scam are similar to those of other types, for example convincing victims of the authenticity of delivered phishing information or using interesting phishing information to collect online credentials. However, phishing scams that are not conveyed through emails are carried out differently. Instead of sending fraudulent and alluring contents, the phishers must initially connect with victims using online communication tools. To send information to specific users, it is necessary to obtain registration details from the web services of online communication tool providers. In this case, phishers either pretend to be acquaintances of the victims or they take advantage of other malware or traditional phishing scams to steal login credentials. Scams that aim to install malware on the personal computers of victims are similar to traditional phishing scams, for example when browsing interested websites where malware is hosted, or infection by installing applications from unknown or un-authenticated publishers. Once malware is installed on a victim's computer, access rights to the account of the victim are granted. Then the victim's credentials can be compromised. Usually, the malware is used to collect the credential information about online banking. Nowadays, depending on the type of the malware, online credentials at the social media site (Chhabra et al. 2011, Hong 2012) can be captured, or the operating systems on the connected devices can be compromised (e.g. key loggers can be installed on the operating system).

Phishing scams through mobile devices

Although deception cases through cell phones have been reported (Ryst 2006, Swartz 2006), most of them spoof victims by recharging pre-paid subscriptions or mislead victims into calling expensive phone services. As cell phones are becoming equipped with more and more computation and communication power, these enhancements are bringing with them more possibilities for phishing scams to be launched. For example, the phone number shown to the victim can be modified into some trusted number such as an emergency call number. Typically users do not know that these emergency call numbers are not normally used for calls. Therefore, end users can easily be cheated believing that the incoming calls are from the public sectors. Once victims are convinced, typical phishing scams can be deployed. According to Chen (2008), phishing cases using mobile devices may be categorized into 12 main types. Even though phishing medium has moved to mobile devices, the targets and contents remain similar to those of traditional phishing scams.

Phishing scams in multimedia

This type of phishing scam is usually conducted via non-email media, e.g., video and audio clips. It appears to be more secure for victims, since these multimedia features provide more evidence for them to distinguish between authentic and fraudulent information. However, Tencent, one of the largest instant-messaging providers in China, has reported that phishers have been able to take advantage of multimedia and convince victims to offer financial support (Tencent Service 2012). In this type of phishing scam, victims may receive a video clip in which their acquaintances appear, but the audio is manipulated for phishing purposes. To make this type of phishing successful, more effort is required to get to know the victims and their relationships with their acquaintances. Thus, the phishing scams require more long-term planning. Most likely, the victims have misused or lost their credentials and personal information on multiple occasions before this type of phishing scam is launched. For example, a phisher probably needs to know the relationship between a victim and the person who the phisher pretends to be. This required information can be collected, for example, via the victim's online communications. However, in order to have access to the communication history records, certain access rights must be misused. In this case, the credential information to provide access to the communication history must have already been stolen by the phisher.

2.2 Online Services

Online services store users' credentials. At the same time, the security and usability vulnerabilities of these online services are being exploited by phishing attackers. In order to design better phishing prevention systems from the perspective of end users, it is first necessary to understand how these vulnerabilities are exploited.

2.2.1 User Login Authentications

Many types of personal identities are presented online, and users of online services must have their details authenticated before they can legitimately access their private information online (Berki and Jäkälä 2009, Jäkälä and Berki 2004). Traditionally, to log in to a web site, users' personal identities are provided at the designated areas (Figure 1). These areas request the account names and passwords of users. Undoubtedly, this method is vulnerable (Cranor and Garfinkel 2004, p.17). If phishers steal this personal information without users' awareness, phishers are then able to log in and abuse the victims' accounts.

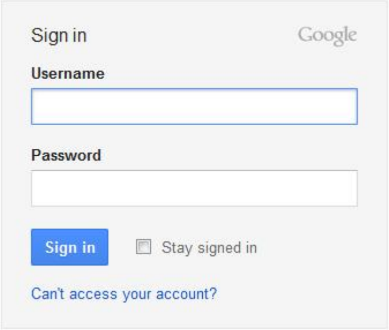


Figure 1. The login area for Gmail users

2.2.2 Non-Login Personal Data for Web Applications

Besides login authentications, many forms of personal data other than login data are given to web service providers. Non-login personal data are usually collected to provide better services. However, users should be worried about the safety of these personal data (Dhillon and Moores 2001). If these personal data are sold to phishers, with detailed personal profile information, phishers are then able to conduct more advanced and successful attacks. Recent reports in particular show that the compromise of online game services by phishing or non-phishing scams can result in the loss of users' personal data, which can be copied and misused by the criminals (Zorz 2011).

2.2.3 Secure Connection Indicators

To indicate a secure online connection, a clearly designed symbol is placed on the main web user interface. To take an example, in the early versions of Internet Explorer and FireFox, i.e., earlier than Internet Explorer 6.0 and FireFox 2.0, a lock icon was placed at the corner of these browsers (Figure 2). When a user clicked on the lock icon, the certificate for the visited web page displayed encryption and issuer information. Though users were able to view this information to make informed decisions on the web page, the visibility and understandability of the lock design was prone to phishing. According to research findings (Balfanz et al. 2011, Li et al. 2012), Secure Sockets Layer (SSL) and Transport Layer Security (TLS) components are important in building secure online communication channels, but for users these security components must be applied effectively in a well-designed application context.

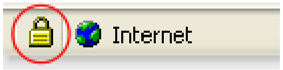
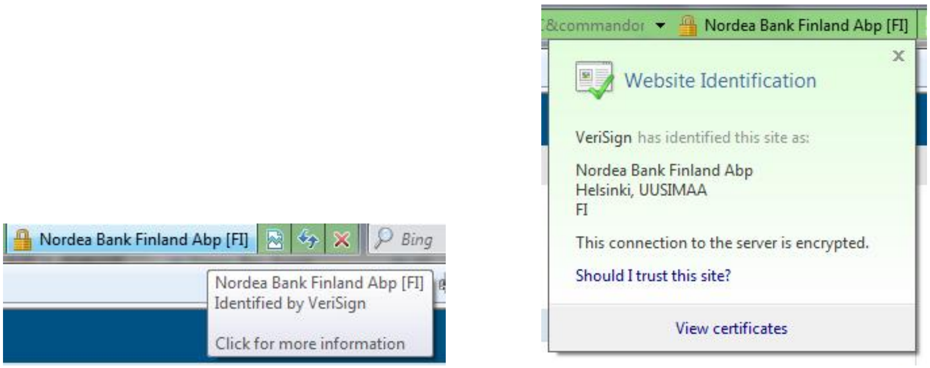


Figure 2. The lock icon at the bottom-right corner of the user interface of Internet Explorer 6.0

In later versions of web browsers, the lock icon became more informative. From Internet Explorer 7.0, the lock icon has been placed on the address bar together with information on the current web page (Figure 3). To notify visitors about the security of web pages, Google Safe Browsing has been embedded into the FireFox browser since version 3.0 (Figure 4). When a malicious web page is detected, Google Safe Browsing stops downloading the malicious web page and instead shows a blocking page. At the same time, the security indicator on the address bar turns red to warn the user (Figure 5).



Nordea Bank Finland Abp [FI]
Identified by VeriSign

Click for more information

Website Identification

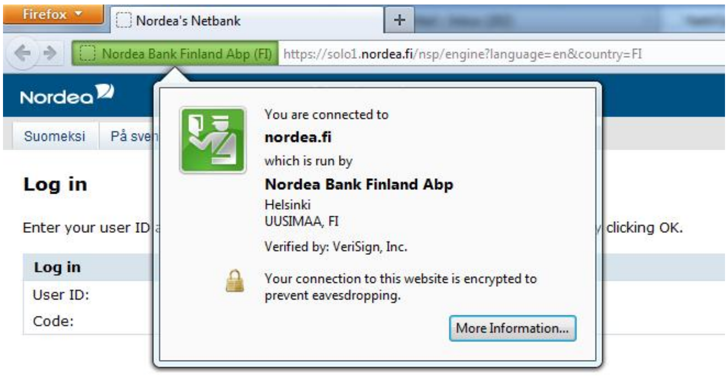
VeriSign has identified this site as:

Nordea Bank Finland Abp
Helsinki, UUSIMAA
FI

This connection to the server is encrypted.

Should I trust this site?

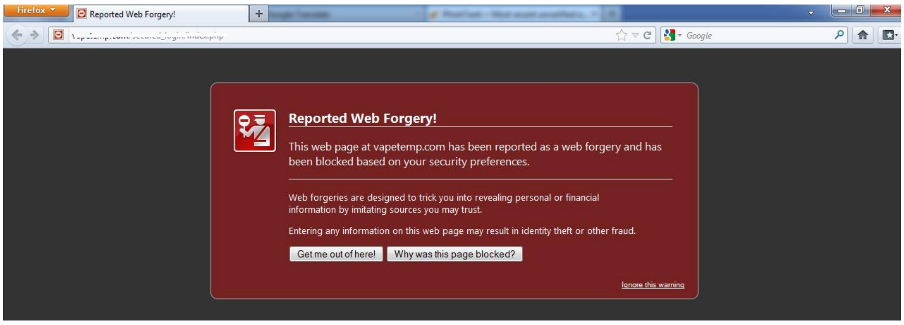
Figure 3. The lock icon on the address bar of Internet Explorer 8.0



You are connected to
nordea.fi
which is run by
Nordea Bank Finland Abp
Helsinki
UUSIMAA, FI
Verified by: VeriSign, Inc.

Your connection to this website is encrypted to prevent eavesdropping.

Figure 4. The web page security indicator provided by Google Safe Browsing on the address bar of Firefox 6.0



Reported Web Forgery!

This web page at vapetemp.com has been reported as a web forgery and has been blocked based on your security preferences.

Web forgeries are designed to trick you into revealing personal or financial information by imitating sources you may trust.

Entering any information on this web page may result in identity theft or other fraud.

Figure 5. Google Safe Browsing (integrated with Firefox 6.0) detects a fraudulent web page

2.2.4 Cookies

Cookies are small sets of state information that record the browsing history and online service preferences of certain websites (Barth 2011). When a user browses a web page or plays a Flash video, the hosting website is able to send state information to the client's browser and retrieve client-side state information. Attackers may apply cross-site scripting (XSS) to steal or misuse cookies. For example, attackers can upload the cross-site scripting codes to an XSS-vulnerable web site. These codes are designed to download and collect the cookies stored at the victim's browser. Once a victim visits the web page with XSS codes, the cookies can be stolen. With stolen cookies, attackers can initiate unauthorized actions on the target website on which victims have established a private connection. For these reasons, cookie management tools were implemented to protect users' privacy by monitoring, rejecting and blocking cookies from websites (Surfthenetsafely 2011a,b, S1tony 2011, Yardley 2007).

2.2.5 Public-Key Infrastructure for Online Services

To prevent eavesdropping and assure the authenticity of both parties on the Internet (clients and servers), web browsers are equipped with public-key infrastructure (PKI) and multiple secure online transmission protocols, including SSL and its successor TLS. In these protocols there are two encryption keys defined and used in one secure transportation event: one public key and one private key. Both parties in PKI-based online communication have their own set of key pairs. Because SSL protocol is vulnerable, e.g. to the man-in-the-middle attack (Turner and Polk 2011), TLS protocol is more and more supported by the security professionals. Since the research scope of this thesis is on the user-centered quality, in this part, the detailed implementation of a secure transportation protocol is not covered. Instead, only the human-computer interaction design of the secure transportation protocol is discussed in this section.

The public keys of web sites are published by third-party organizations and pre-stored to the caches of web browsers. When a user visits a web site using a SSL or TLS connection, several steps are prepared before the actual communication starts, e.g., handshake and key pair exchanges (Figure 6). Taking the design of the TLS protocol version 1.2 (Dierks and Rescorla 2008) as an example, the first step is handshake acknowledgement, which is the same as the handshake step of other regular Transmission Control Protocol (TCP) communications except that additional information such as the secure transportation protocol version and the cipher suite is included. After that, the client-side browser generates the client’s key pair and transfers the encrypted client’s public key to the server side. Then, a master secret is combined from a random number and the client’s public key to generate all other key data. Then a *Finished* message containing a message authentication code (MAC) over the previous handshake message is sent between the client and server side. When the server and the client both verify the contents in the *Finished* message, they can start to use the pre-defined content type, either encrypted or plain-text, for later secure communication.

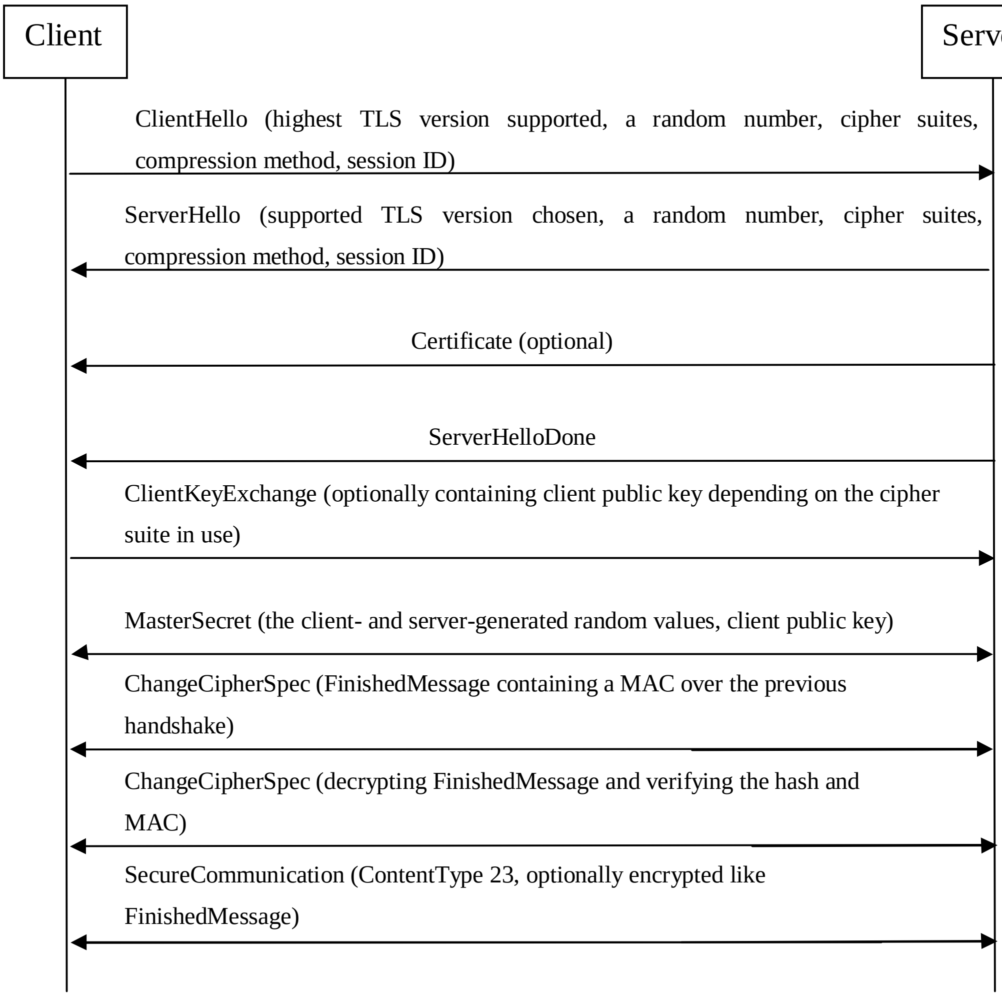

Client
Server
ClientHello (highest TLS version supported, a random number, cipher suites, compression method, session ID)
ServerHello (supported TLS version chosen, a random number, cipher suites, compression method, session ID)
Certificate (optional)
ServerHelloDone
ClientKeyExchange (optionally containing client public key depending on the cipher suite in use)
MasterSecret (the client- and server-generated random values, client public key)
ChangeCipherSpec (FinishedMessage containing a MAC over the previous handshake)
ChangeCipherSpec (decrypting FinishedMessage and verifying the hash and MAC)
SecureCommunication (ContentType 23, optionally encrypted like FinishedMessage)

Figure 6. A TLS handshake example (Dierks and Rescorla 2008, Brown and Housley 2010, Wikipedia 2012)

From the example of the TLS handshake, it is clear that the design is accomplished from the point of view of service providers. For instance, web browsers do not warn users if a secure transaction protocol is not used at a web site. In addition, the user interface (e.g. Figure 2) of secure connections is not explicit and informative to prevent from phishing attacks (Li et al. 2012). In this case, if a server is not an authentic online service, the user is not informed about this during the handshake process. Therefore, although the design helps to verify the authenticity of both parties in communications, it is still vulnerable to man-in-the-middle attacks (Asokan et al. 2003, p.33) and phishing attacks because of this lack of usability from the perspective of a user attempting to identify the authenticity of an online service (Garfinkel 2003, p.21).

2.3 Anti-phishing Software

In this section, examples of popular anti-phishing software are described. Phishing prevention systems are categorized, and their current advantages and disadvantages are reviewed.

2.3.1 Client-Side Anti-phishing Software

Examples of anti-phishing applications on the client side are SpoofGuard (2001), Netcraft (2005), Google Safe Browsing (2006), DOMAntiPhish (Rosiello et al. 2007) and Anti-phishing IEPlug (2007). The main technologies in these applications are text- or image-based content analysis (SpoofGuard), blacklist-based Uniform Resource Locator (URL) analysis (Netcraft, Google Safe Browsing) and whitelist-based URL analysis (DOMAntiPhish, Anti-phishing IEPlug). Text- or image-based analysis is self-adaptive in order to analyze content automatically. A text-based self-adaptive filtering method is designed to classify the emails according to a trained algorithm. To train the algorithm, it is necessary to prepare a set of test samples, known as a corpus. These test samples are collected from real life and accurately classified before the training starts. Sometimes, these corpora are pre-processed, e.g., by tokenization, which means to turn a sample document into a set of words, or by lemmatization, which uniforms the different tenses or forms of words. When all the words are tokenized and lemmatized, it is time to extract features, which means looking for the most relevant terms in a document. Examples of feature-extracting methods include information gain and mutual information (Zhang et al. 2004). After that, the processed corpora and their classification are given to the filtering algorithm. There are many classification algorithms for text filtering. These algorithms look for the terms most likely to appear in the known categories, which are defined and trained with training corpora. However, it is easy to misuse this self-adaptive analysis capability if the software is not well trained (Fawcett 2003). For example, phishers are able to use different languages, alternative words, or the news topics to circumvent content filtering.

The quality issues for blacklist-based URL analysis are (i) how frequently the list is updated and (ii) the quality of the list, including false positives or negatives and extensiveness. To maintain such a blacklist, a certain amount of human resources are also needed to manually evaluate reported suspected URLs and to reliably and efficiently release updated versions to end users. Whitelist-based detection is ideal for personalization, but the phishing detection rate can be improved by assistance from other content analysis algorithms. If the phishing detection is based on the Document Object Model (DOM) structure of the given URL web pages, it can easily be bypassed by using certain web page design techniques such as JavaScript.

In addition to the use of phishing content-filtering applications, access control on personal computers has been strengthened on the client side. Since its Windows Vista operating system, Microsoft (2011) has provided upgrades to its User Account Control (UAC) to help users manage access rights by elevating privileges. When an application attempts to perform critical system changes at application level, e.g., installation, un-installation, registry entry changes, driver changes and system variable changes, the UAC checks the integrity and access privileges of the application. If a user grants administrative privilege to the application, the changes take place. Otherwise, the system change request is discarded. The UAC feature is able to detect application publishers who are not registered and verified by Microsoft. On mobile devices, trust computing also takes place. The Mobile Trust Module 2.0 specification supported by multiple chip vendors is under development, which may help to define a trusted module to monitor the authorization and integrity of software deployed on mobile devices (Trusted Computing Group 2011a,b).

In addition to content filtering and upgrades to the UAC system, it is important to prevent the installation of malicious programs on personal computers. For example, on the client side, the “Hax door” malware (Haxdoor, 2011), using rootkit technology (Rootkit, 2011), is able to overwrite the original master boot record and unload critical system binaries or services to silently “hook” the system messages, redirect URL requests or record personal data. In this case, it is necessary to equip personal computers with superior anti-virus or anti-spyware software together with client-side anti-phishing software.

2.3.2 Server-Side Anti-phishing Software

An example of anti-phishing and anti-spam software on the server side is SpamAssassin (2011). It was the first content classification feature to use naive

Bayesian classification to filter and detect untrustworthy email contents. The Bayesian algorithm in SpamAssassin compares the frequency of terms between the email being checked and previously detected malicious emails. If the comparison shows that the terms appearing in the given email are very similar to those in the detected malicious emails, the checked email is marked as spam or phishing. Commtouch uses another technology to detect spam and phishing messages, called Recurrent Pattern Detection (Commtouch 2011). Instead of checking term-appearance frequency, this technology learns the message distribution and structure patterns from message envelope, subjects, and so on. When malicious patterns are collected from Internet traffic, the Commtouch center releases the collected patterns to individual clients to detect malicious emails.

To verify the authenticity of received emails, Microsoft SenderID (2011) and Open DomainKeys Identified Mail (2011) were developed. In general, if a mail transfer agent receives an email, it will send a DNS request to a dedicated DNS server where the trusted domains and IP addresses are stored. If the registered IP address matches the sender’s IP address in the email, it is verified as being from an authentic email sender. Otherwise, the email is indicated as a non-authenticated one.

Original DNS protocols are vulnerable to DNS poisoning, which is to send malicious DNS responses to the resolver and pollute the cache of the resolver. In addition to DNS poisoning, spoofing among autonomous systems also occurs (Donnerhacke 2011, Israr et al. 2009). DNS poisoning is always employed by phishers in pharming attacks (Helenius 2006). Therefore, it is necessary to consider how to stop phishing that uses DNS poisoning techniques. Domain Name System Security Extensions (DNSSEC 2011) is an infrastructure that is able to prevent DNS poisoning and similar spoofing among autonomous systems. In DNSSEC, each DNS lookup record is signed by the private key of a DNS server that responds to the DNS lookup. If the signed DNS lookup record can be authenticated with the public key of the DNS server, it proves that the DNS record is from an authentic DNS server. DNS is a hierarchical distributed naming system. Therefore, DNSSEC must build on a chain of trust in order to authenticate the public key of each DNS server.

The root zone is managed jointly by the U.S. Department of Commerce, the Internet Corporation for Assigned Names and Numbers (ICANN, a private non-profit organization to coordinate the global Internet’s systems of unique identifiers) and VeriSign (a commercial organization providing online security solutions).

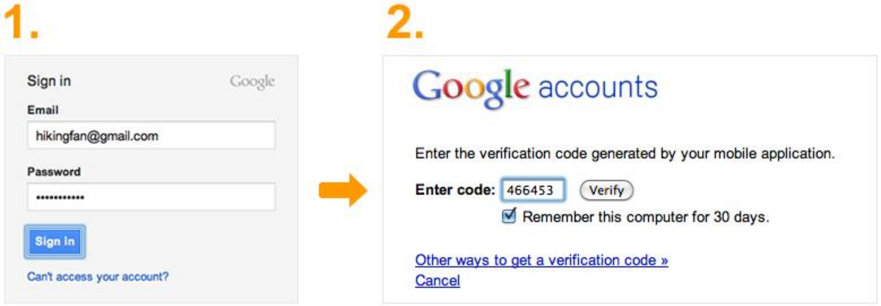
ICANN is responsible for vetting and processing changes to a Delegation Signer record. Then, the Department of Commerce authorizes the changes to the root zone. After that, VeriSign signs the new root zone and distributes the signed root zone to the root servers. ICANN updates and publishes the Key Signing Key sets.

The public key of each DNS server is authenticated by locally calculating the hash record of the public key of the DNS server where the DNS data is sent from, and comparing this with the hash record in the DNS response. If these two hash records are the same, it is proved to be authentic.

Another phishing-resistant design is OAuth (2011). This is a public protocol designed to guarantee the authenticity of both parties during communication. An authentication certificate is issued by a third party that is believed to be trustworthy by both of other parties. OpenID is another standard that supports inter-authentication among different parties in communication (Bellamy-McIntyre et al. 2011). Unlike OAuth, the ID issued through OpenID by the trusted third party is in plain text rather than encrypted data. Therefore, an extension was created for OpenID designed for phishing resistance.

To offer more phishing-resistant login authentication services, various new and more secure login methods have been introduced. In general, these upgraded methods focus on *What you know* and *What you have* factors (Helenius 2006). For example, Gmail has been upgraded to use a two-step authentication login method (Figure 7), which sends a verification code from the Gmail server to a user-specified mobile device via Short Message Service or generates a verification code via a dedicated Google application on the mobile device in order to confirm the login authentication (Gmail Blog 2011). This method offers users more channels to identify themselves and confirm their login activities.

Another example of an upgraded login method is one-time user identity. This method is usually employed by online banking or financial web services. To take the Nordea Bank in Finland as an example, its customers can log in to its online banking services only by providing a user ID and access code. The access code is a four-digit random number printed on a letter issued by Nordea through the post, and on each letter there are 80 access codes and 18 four-digit codes to confirm every transaction request. Similar to Nordea Bank, PayPal uses a two-factor authentication method, the security key (Figure 8), which generates a six-digit password for users to log in to its online services.



Sign in

Email: hikingfan@gmail.com

Password: ••••••••••

Can't access your account?

Google accounts

Enter the verification code generated by your mobile application.

Enter code: 466453

☑ Remember this computer for 30 days.

Other ways to get a verification code »

Cancel

Figure 7. Two-step authentication in Gmail (cited in Gmail Blog 2011)

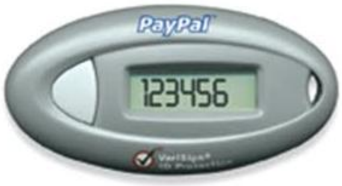


Figure 8. The security key for logging in to PayPal

Besides the security key used by PayPal, many security devices have been applied in various areas for authentication. For example, biometric security devices such as voice patterns and fingerprints are also used in authentication. Although biometrics are not inherently a usable form of security (Coventry 2005), some biometrics offer more authentication factors in a more usable way (than the traditional username–password pair) in order to strengthen the security of existing knowledge-based and token-based authentication systems (Wang et al. 2008, Ngugi et al. 2011, Sae-Bae et al. 2012). Furthermore, when smart cell phones equip more computing power, some mobile security mechanisms were also carried out. For example, Tang and his colleagues (2003) introduced their PIN verification design to improve the security of mobile devices. Concerning security and usability, mobile devices are also used. Mazhelis and his colleagues (2005) presented an identity verification system for mobile terminals which is focus on high verification accuracy, continuous security and usability. Tamrakar and his colleagues (2011) also proposed a design to verify the identities for public transport ticketing systems using NFC-capable smart cell phones.

Nowadays online social media get increasingly popular, and it is also abused as a phishing tool (ZoneAlarm 2012). This attracted attention from various researchers. Chhabra and other researchers (2011) discovered that phishers prefer to use short

URLs on the online social media for phishing. Social snapshots were used to harvest social information from Facebook (Huber et al. 2011). Semenov and his colleagues (2011b) designed an information system to monitor and analyze the information on the social network. This system is able to long-term monitoring of diverse social networks at different social media sites. These are valuable research findings and tools to prevent personal social information to be abused. However, it is still needed to study how to employ these tools for the design of a usable and secure phishing-resistant information system for end users.

After reviewing the different designs of anti-phishing software on both client side and server side, it is beneficial to have a holistic picture of the current state-of-the-art of anti-phishing software. Understanding this current situation is also advantageous for knowing how prevention techniques can be improved to protect users from the emerging and increasingly sophisticated phishing attacks.

3. Software Quality Assurance and Anti-phishing

Various phishing prevention systems have been designed and developed. However, their quality varies, and software quality assurance methodologies have therefore been applied for these phishing prevention systems. This chapter describes the contingency methodologies (particularly the Multiview framework), the standards of software quality assurance and the theories of software usability. Software quality and its management is a wide research area, but this chapter focuses on the software quality methodologies that have been applied in the author's doctoral research and studies. In addition the chapter reviews existing software quality research findings in the field of anti-phishing.

3.1 Contingency Approach and Anti-phishing

In this section, the history of contingency approach is reviewed, and the practicability of using a contingency approach in the anti-phishing research domain is addressed.

Contingency theory was first introduced in the field of economics. Fiedler's (1963) contingency model was created to help management in an uncertain and dynamic industrial and organizational psychology context, by analyzing leadership styles and behaviors. For example, it examines how a company can be managed when managers face a variety of employees. According to the theory, it is not efficient and effective to manage all types of employees in the same way.

This method has also been introduced in the area of software engineering. Before the contingency approach, information system development was carried out with technology-oriented approaches (e.g., Daniels and Yeates, 1971). With these approaches, information system development was conducted by a formal system analysis process, which caused problems and errors in practice (Avison and Wood-

Harper 2003, p.6). For example, the human, social and organizational issues are not thoroughly addressed in many of the information systems research and development methodologies. To solve these practical problems, and in order to place more emphasis on human and organizational aspects, contingency approaches were introduced (Naumann et al. 1980, Avison 1990, Tuunanen et al. 2007).

Multiview is a contingency approach that analyzes information systems in five stages: human activity, information, socio-technical, human–computer interface and technical aspects. At each stage, different development tools, techniques and research methods are employed.

In the first stage, human activity is analyzed by observing the organization. In this way, world views can be identified in order to form the basis of the system requirements and to understand what the information system will be and what it will do.

In the second stage, the information involved in the targeted information system is analyzed. The information, i.e., the entities and functions of the problem situation, is analyzed in two phases. One phase is the development of a functional model to identify the main function and decompose the main function into sub-functions and create data flow diagrams. The second phase is the development of an entity model to define entities and relationships among entities.

In the third stage, socio-technical analysis and design is conducted to take into account both people and their needs together with the computer systems and necessary work tasks. This analysis and design includes choosing systems, socio-technical alternatives, and creating requirements for the computers, the personal working process and the socio-technical roles.

In the fourth stage, the human–computer interface is designed. This is concerned with how users will interact with the computer, e.g., its screens, inputs and outputs.

In the fifth stage, technical aspects are analyzed with the aim of achieving an efficient design that meets the given systems specifications. One method to conduct technical aspect analysis is to break down the system into sub-systems, e.g., application subsystem, information retrieval subsystem, database subsystem, control subsystem, control subsystem, recovery subsystem and monitoring subsystem.

By investigating and analyzing the information system from these five aspects, hidden and implicit requirements in uncertain circumstances can be elicited in order to facilitate the design of the system and improve system quality.

Phishing is a dynamic threat, meaning that phishing scams are continually increasing in sophistication. For example, new tools were recently created to automatically generate phishing web sites (Evers 2007); the online video chat was even more recently abused by phishers (Tencent Service 2012). These events and trends indicate the dynamic evolution of phishing. Thus, if anti-phishing software is currently designed to filter out the phishing content in the future emails, it will be useless when more future sophisticated phishing attacks are launched. To deal with dynamic phishing problems, it is not possible to create anti-phishing software to prevent certain type of phishing attacks, but it is necessary to analyze the requirements from different perspectives and from different systems (e.g. email system, online social media, and information systems). In this regard, a contingency approach is feasible for analyzing a phishing-resistant system combining multifaceted aspects of anti-phishing activities.

3.2 Software Quality Assurance

Software quality metrics derived before the software is delivered provide a quantitative basis for making design and testing decisions (Pressman 2001, p.81). In order to assure the user-centered quality for anti-phishing software, it is needed to have a reliable reference on how to address and consider software quality metrics by other researchers. For example, Pressman (2001, p.96) suggested four measures of software quality, including correctness, maintainability, integrity, and usability. Correctness means that the software operates correctly. Maintainability is the ease with which a program can be corrected if an error is encountered, adapted if its environment changes, or enhanced if the customer desires a change in requirements. Integrity is to measure how the software can prevent from certain security attacks and threats. Usability considers four characteristics, including learnability, efficiency, productivity and users’ subjective attitudes.

Another well-known reference is the international standard, ISO-25010. The quality model in the ISO-25010 (2011) standard includes eight categories: functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability and portability. Functional suitability focuses on software’s functions and properties that meet stakeholders’ requirements. Performance efficiency is concerned with the relationship between the level of performance and the amount of the used resource of the software under the required conditions. Compatibility means that when the environment changes, the software is still able to be executed as desired. Usability addresses how easy the software is to use. For example, can users recognize the software? Can users learn how to use the software easily? Reliability makes sure that the required software works in a consistent manner to maintain its level of performance. Security guarantees that the data and the software are accessible only to the authorized. Maintainability refers to the required efforts when some specified modifications are needed. Portability is concerned with whether the software can be deployed and transferred to the required environments.

Besides the above quality model, ISO-25010 defines quality-in-use metrics. There are five sub-classes in quality-in-use metrics: effectiveness, efficiency, satisfaction, freedom from risk, and context coverage. These quality-in-use metrics are quality measurements when the final product is used in real conditions. Therefore, they are useful in analyzing the quality of anti-phishing applications and phishing-resistant systems because of the nature of security requirements (Sindre and Opdahl 2001, Li et al. 2007).

In order to help prevent phishing attacks, the present research used three methodologies to collect the appropriate requirements, including misuse-case-based requirement elicitation, and two types of software usability evaluation. In the later sections of this chapter, the theories and practices of misuse-case-based security requirement elicitation, software usability evaluation and user requirements for the performance of text content-filtering algorithms are introduced.

3.3 Misuse-Case-Based Security Requirement Elicitation

Pressman lists the software quality metrics to be measured Pressman (2001, p.96), but misuse cases are a method of eliciting security requirements. For the Internet threats such as phishing attacks that take advantage of vulnerabilities in security architecture design, misuse cases can help to improve the quality of phishing-resistant system design.

The first misuse case test was invented by Sindre and Opdahl (2001), with the aim of testing and evaluating security breaches in design and use cases at the requirements stage. In general, a misuse case is an inverse use case in which the user requirements are defined. Since misuse cases are generated by determining the vulnerabilities in use cases, use cases are required as inputs.

Similar to use cases, in misuse case specifications, certain fields must be specified. The key fields include basic paths, alternative paths, capture points and extension points. Basic path is the primary method in which a criminal is able to abuse a certain use case. Alternative paths list further potential methods to misuse a use case. Capture points describe how the misuse can be potentially stopped or detected. Extension points present other possible misuse cases that may be taken advantage of by the criminal.

Besides these key fields, other fields describe the roles, tracing information, rationales and the threat consequences of misuse cases, e.g., trigger, preconditions and related business rules. In addition, descriptive fields are included in misuse case specifications, such as misuse case scope, profile, level and stakeholders. A sample misuse case template (Sindre and Opdahl 2001) is given in Tables 3 and 4. The security requirement in this sample is that the system is able to protect the password from being obtained by the criminals from the Internet or the computers which may be infected by malware.

Table 3. Misuse case description, part 1

Name: Obtain Password
Summary: A criminal obtains and later misuses operator passwords for the e-shop by tapping messages sent through a compromised network host during operator logon.
Author: David Jones
Date: 2001.02.23.
Basic path:
bp0 A criminal has hacked a network host computer and installed an IP packet sniffer (step bp0-1).
All sequences of messages sent through the compromised host and that contain strings such as 'Logon', 'User name', 'Password', 'passwd' etc. are intercepted and analyzed further (step bp0-2 and extension point e1).
In this way, the criminal collects (likely) usernames and passwords along with the IP addresses of the computers they are valid on (step bp0-3). The criminal – possibly much later – uses the username and password to gain illegal operator access to the e-shop computer (step bp0-4).
Alternative paths:
ap1 The criminal has operator privileges on the network host. No hacking of the network computer is necessary (changes step bp0-1).
ap2 The criminal has not penetrated a network host, but instead intercepts messages sent through the telephone system from the e-shop operator's home (changes step bp0-1).
ap3 Instead of home phone, the criminal intercepts messages sent from the e-shop operator's portable devices (changes step bp0-1).

Capture points:
cp1 The password does not work because it has been changed (in step bp0-4).
cp2 The password does not work because it is time dependent (in step bp0-4).
cp3 The password does not work because it is different for different IP addresses (in step bp0-4).
cp4 Operator logon to the e-shop is only possible from certain IP addresses (in step bp0-4).
cp5 Communication tapping (in step bp0-2) is not possible (perhaps because the communication is encrypted).
Extension points:
ep1 Includes misuse case "Tap communication" (in step bp0-2).

Table 4. Misuse case description, part 2

Triggers: tr1 Always true, i.e., this can happen at any time.
Preconditions:
pc1 The system has a special user 'operator' with extended authorities.
pc2 The system allows the operator to log on over the network.
Assumptions:
as1 The operator uses the network to log on to the system as operator (for all paths).
as2 The operator uses his home phone line to log on to the system as operator (for ap2).
as3 The operator uses his home phone line to log on to the system as operator (for ap3).
Worst case threat (postcondition):
wc1 The criminal has operator authorities on the e-shop system for an unlimited time, i.e., she is never caught.
Capture guarantee (postcondition):
cg1 The criminal never acquires operator authorities on the e-shop system.
Related business rules:
br1 The role of e-shop system operator shall give full privileges on the e-shop system, the e-shop system computer and the associated local network host computers.
br2 Only the role of e-shop system operator shall give the privileges mentioned in br1.
Potential misuser profile: Highly skilled, potentially host administrator with criminal intent.
Stakeholders and threats:
sh1 e-shop

- reduced turnover if misuser uses operator access to sabotage system
- lost confidence if security problems get publicized (which may also be the misuser's intent)

sh2 customer

- loss of privacy if misuser uses operator access to find out about customer's shopping habits
- potential economic loss if misuser uses operator access to find credit card numbers

Scope: Entire business and business environment.
Abstraction level: Misuser goal. **Precision level:** Focused.

Misuse cases are able to list potential security breaches and vulnerabilities in use cases. In this way, designers and stakeholders can explicitly determine the security requirements.

3.4 Software Usability and Anti-phishing

According to the definition in ISO-9241-11 (2010), usability emphasizes that a specific user is able to complete specific goals in a specific context of use from three perspectives: *efficiency*, *effectiveness* and *subjective satisfaction*. These three metrics describe what can improve the software's usability. When measuring *efficiency*, more detailed metrics are used, such as:

1. How much time does it take for a user to complete a task?
2. How much time does it take to learn to use the software?
3. How much time does it take to recover the software from errors?
4. How many errors are there in the software?
5. How frequently do users consult manuals or help pages?
6. How frequently are the commands repeated or unsuccessful?

To measure *effectiveness*, the accuracy and the percentage of completed tasks in a given period of time is calibrated. For example:

1. How many tasks are completed?
2. How many tasks are completed correctly?

The *satisfaction* is very subjective. Therefore, it is common to conduct surveys or interviews with questions such as:

1. Does the user prefer the functionalities offered by the software?
2. Does the user enjoy using the product?
3. How often does the user experience negative feelings?

In the definition given by Jakob Nielsen (1993), usability, as an attribute of system acceptability, consists of five sub-attributes, including learnability, efficient to use, easy to remember, few errors and subjectively pleasing. Learnability refers to the metrics to measure the time it takes for a new user to attain a certain level of performance with the software. Efficient to use metrics are similar to *efficiency* defined in ISO-9241-11. Memorability means how easily users are able to remember the operations given by the software. Ideally, there should be no need to relearn how to use the software every time it is used. Few errors mean that the operation as a whole flows smoothly. The definition of subjectively pleasing is similar to that in ISO-9241-11 and is recommended for questionnaires, interviews and psycho-physiological measures.

No matter which usability metrics are used, the users and the context should be taken into account. To study users, usability researchers determine targeted users and their characteristics. With a correct understanding of the software context, researchers can narrow down and simplify the scenarios and facilitate the simulation of these scenarios in the laboratory for usability evaluations and tests.

Two of the most popular usability evaluation methods are heuristic evaluation and usability testing. Heuristic evaluation checks usability problems in software according to a heuristic list where usability is defined. Nielsen (1993) specifies 10 heuristics as follows:

1. Visibility of system status (visibility)
2. Match between system and the real world (familiarity)
3. User control and freedom (freedom of choice)
4. Consistency and standards (consistency)
5. Error prevention (error prevention)
6. Recognition rather than recall (recognition)
7. Flexibility and efficiency of use (flexibility)
8. Aesthetic and minimalist design (aesthetics)
9. Help users recognize, diagnose and recover from errors (error messages)
10. Help and documentation (help)

To evaluate software usability from the perspective of end users, usability tests and corresponding tasks are prepared to observe the level of usability of the tested software. Although metrics of usability are defined in ISO-9241 (2010) and by Nielsen (1993) for different software, the tasks vary. To guarantee the design quality of a usability test, a pilot test usually needs to be performed. During the pilot test, the prepared test tasks are run through by invited participants. The usability test design should be improved if any test design problem is found during the pilot test. After the usability test tasks have been prepared, participants are asked to complete the tasks and keep thinking aloud. Thinking aloud means that the participants must keep talking about what they are doing and what they are concerned about when completing the given usability test tasks. During the usability test, all of the users' activities are recorded for further analysis.

For anti-phishing toolbars, different usability metrics are used. According to research findings (Wu et al. 2006b), users' primary goal is to log in to an online web service when they have handed over their credentials instead of having to first identify the authenticity of the visited web page. However, to evaluate the usability of anti-phishing toolbars, usability test participants must try out the functionalities of the tested toolbars (Egelman et al. 2008, Jakobsson and Ratkiewicz 2006, Wu et al. 2006a,b). In these usability studies, participants were asked to complete a set of prepared tasks (e.g., determine the authenticity of the given web pages), and the activities of these participants were observed and recorded. By observing the activities of the participants, the researchers were able to compare the usability of the tested anti-phishing applications.

Piazzalunga et al. (2005) conducted a usability evaluation to compare new security devices. In their research, they defined usability metrics, including security errors, hesitations and mobility errors. After the experiment, the researchers suggested general recommendations on the design of usable security devices:

1. The system as a whole matters
2. Devices that require a reader affect mobility
3. Add value to the security device
4. Simple is secure
5. Tune and adapt the software
6. Conduct experimental usability studies, even small ones

Although both heuristic evaluation and usability end-user tests have been used, they have their own advantages and disadvantages. A comparison of these is listed in Table 5.

Table 5. Comparison of usability evaluation methods in anti-phishing research

	Advantages	*Disadvantages*
Heuristic evaluation	1. It is easy to prepare the environment, e.g., phishing attack simulation. 2. It is easy to define heuristics for anti-phishing applications. 3. It does not take as much time as a	1. No real end users are involved.

	usability test. 4. Several products can be tested while fewer participants needed.	
Usability test	1. Real end users are involved. 2. It is easy to measure the results. 3. The test environment is similar to real situations in the phishing context.	1. There are limited number of different types of phishing attacks can be included. 2. There are more ethical issues to be considered when designing the test tasks. For example, participants' privacy should not be exposed during the test; phishing attacks used in the lab must be isolated; and participants should not be embarrassed during the test.

3.5 Performance Experiments and Anti-phishing

To filter malicious messages, self-adaptive filtering algorithms were introduced into anti-phishing solutions. To evaluate the performance of these filtering features, metrics were prepared (Yang and Pedersen 1997, Sahami et al. 1998, Androutsopoulos et al. 2000a,b, Zhang et al. 2004, Webb et al. 2005, Comack and Lynam 2007, Youn and McLeod 2007, Comac and Kolcz 2009), including false-positive rate, false-negative rate and total cost ratio. False positive means that legitimate messages are misjudged as malicious, and false negative means that spam/phishing messages are misclassified as legitimate. The total cost ratio (TCR) metric (Androutsopoulos et al. 2000b) compares the cost of spending on manually deleting spam messages without any spam filter and the time to process false-positive and false-negative emails, which can be simply expressed as follows:

$$TCR = \frac{N_{spam/\,phishing}}{\lambda \cdot n_{legit->spam/\,phishing} + n_{spam/\,phishing->legit}}$$

Misclassification cost (λ) above indicates the cost of *false positive* and *false negative* in the experiment. The authors (Androutsopoulos et al. 2000b) seem to want to emphasize that false positives are more important than false negatives for the users.

Results of comparisons (Yang and Pedersen 1997, Sahami et al. 1998, Androutsopoulos et al. 2000a,b, Zhang et al. 2004, Webb et al. 2005, Comack and Lynam 2007, Youn and McLeod 2007, Comac and Kolcz 2009) showed that the performance of the Bayesian algorithm was good. After sufficient training, the algorithm was able to correctly detect approximately 90% of tested spam text email samples with the highest TCR values in the tests (when λ=1, TCR=5.41; when λ=9, TCR=3.82; and when λ=999, TCR=2.86).

4. Author’s Contributions

This chapter describes and discusses the author’s contributions to the field of user-centered quality assurance for anti-phishing software and phishing-resistant systems. Based on my contributions and my work on this topic, seven papers were published and used as an important part of the present thesis. In the following sections, I introduce each paper and outline their contributions to the current research topic. This chapter is divided into two main parts. The first describes two papers related to misuse cases and performance research undertaken to improve the quality of anti-phishing applications and phishing-resistant system design. The second part of the chapter describes five papers that explore how to improve the quality of phishing prevention and phishing-resistant systems according to findings from end-user investigations.

4.1 Misuse Cases and Performance Research to Assure the Quality of Anti-phishing Applications and Phishing-Resistant System Design

To assure the quality of phishing prevention and phishing-resistant software, it is imperative to guarantee that their design fulfills the requirements of users. Therefore, I aimed to find a reliable way to elicit or collect user requirements from analyses by security experts and relevant support such as misuse cases. A misuse case method is a way of verifying the design to determine vulnerabilities from use cases (Sindre and Opdahl 2001). These methods are useful in eliciting security requirements. However, it is not known whether they are also suitable in the design of high-quality anti-phishing software and phishing-resistant systems, particularly as phishing scams are becoming increasingly advanced. In this context, my contribution was in the form of research into how misuse cases could be used to design phishing-resistant information systems.

Phishing content filtering is an important feature of phishing prevention. In order to assure the quality of phishing prevention, it is essential to look at how the performance of phishing content filtering can be guaranteed. Since phishing content filtering also uses adaptive algorithms, I conducted a literature review study to discover how to design a reliable performance experiment for phishing prevention. In this way, the performance quality of phishing prevention can be measured. In addition, elements in the design of a test for reliable performance may be a useful reference for improving the quality of next-generation phishing-resistant systems.

4.1.1 Paper 1 – Misuse Cases to Assure the Design Quality of Phishing-Resistant Information Systems

Software quality features and functionalities such as suitability, accuracy, security and interoperability should be dealt with at the initial stages of software development (Berki and Georgiadou 1996, Georgiadou et al. 2003, Berki et al. 2004, Berki 2006). However, these features and functionalities are not always carefully designed no matter what software development process is applied (Berki et al. 2004, Berki 2006). Phishing, as an online threat combining social engineering and security attacks, takes advantage of security design flaws. To help investigate such security design flaws, we decided to evaluate an example, an online music purchase system, to determine their vulnerabilities to phishing in misuse cases.

To apply a misuse case phishing prevention method, we followed the instructions given by Sindre and Opdahl (2001). In order to verify our proposed method, we presented an example, the MusicBox online service, to demonstrate how the phishing-prone vulnerabilities can be detected and how their countermeasures can be applied.

In this research, we designed with the help of the misuse case methodology a phishing-resistant system, an online music purchase system. In this paper, we started one misuse case and followed instructions to further elicit security requirements. We observed the whole process and concluded that the security requirements for phishing-resistant design could be collected with misuse case methodology. To improve the quality of a phishing-resistant information system, it may therefore be helpful to collect as many such misuse cases as possible. However, we also realized that it was important to (i) define further software quality criteria for a design architecture that contributes to phishing prevention systems and (ii) make misuse cases more valuable (and perhaps more reusable) and more formal in order to be used as a quality assurance technique in the validation and verification of a system.

4.1.2 Paper 2 – Design Reliable Performance Experiments for Text-Content-Filtering Algorithms

In addition to user interfaces, the logics and algorithms of anti-phishing software should be examined when researching the quality assurance of phishing prevention. In order to discuss quality issues in the algorithms applied by anti-phishing software, we focused on methods of designing a reliable performance experiment for different algorithms in text-content filtering.

Using a literature review research method (Järvinen 2010), we examined how other researchers evaluated the performance of text-content-filtering algorithms and the reliability of their performance design. First we carried out preliminary research into the publications in this area and selected representative publications according to their popularity and relevance to our research.

After an analysis and literature review of these representative publications, we found that performance evaluation experiments have similar problems in their design, particularly for machine-learning-based content-filtering algorithms. The most critical deficiency is the lack of work conducted on corpora. Firstly, different languages may impact the performance of classifiers (Zhang et al. 2004). Secondly, noise, e.g., camouflaged messages (Webb et al. 2005), could significantly impact the performance of classifiers. Therefore, well-grounded research on corpora design is urgently needed, particularly on introducing noise into corpora.

We conducted this literature review in order to discover the strengths, weaknesses and limitations of many performance evaluation experiments of email content-filtering methods. This study constituted a natural part of my thesis research on establishing software design quality criteria for anti-phishing/anti-spam software technologies, with the expectation that these criteria may be useful for software application users and software developers.

Inevitably, spam/phishing content filters will continue utilizing and applying artificial intelligence and machine-learning capabilities. In order to evaluate the performance of these different technologies, it is necessary to establish unbiased techniques. This means that the adopted evaluation method should not focus merely on feature size, misclassification cost and filtering algorithms. It is equally important to focus on the diversity of the samples, e.g., how spam filters perform when noise is added to email samples, or how the different sizes and types of attachments affect the accuracy and speed of the process.

4.2 Improving Quality of Anti-phishing and Phishing-Resistant Systems Based on End-User Studies

Phishing attacks, as a type of social-engineering attack, are not simply able to steal private personal information, but can also negatively impact people's personal and professional lives. To understand how users' lives are disrupted by phishing, we invited end users from various countries and conducted a survey to collect their opinions on phishing emails.

To evaluate the variety of anti-phishing software and determine their advantages and disadvantages, it was essential to conduct usability experiments to collect opinions from end users. In my research, heuristic evaluation and usability testing were used.

Although we understand how users interpret phishing contents and anti-phishing software, it is vital to determine how users act when confronted by phishing scams. Karvonen and Parkkinen (2001) give advice on how to design trustworthy web pages. Shahriar and Zulkernine (2010) attempted to establish potential patterns or models of anti-phishing software to investigate how well the tested anti-phishing software was able to protect end users. Other researchers studied how users collected knowledge and how they employed this knowledge to make decisions on the authenticity of certain phishing content (Dong et al. 2008, Kumaraguru et al. 2006). Although these studies have improved the understanding of phishing and the cause of successful phishing attacks, they are insufficient to assure the quality of anti-phishing software and phishing-resistant systems. Therefore it was important to study user behavior when confronted with phishing scams and to model this behavior with a more abstract and precise methodology. In order to aid the design of a usable and secure authentication mechanism, we also deduced and proposed a new set of usability metrics based on an analysis of existing representative authentication mechanisms and instances of online identity theft.

4.2.1 Paper 3 – How Users Are Disturbed by Phishing Emails

For different types of end users, distinct criteria for each individual user are employed to identify authentic and forged web contents. However, these distinctions may bring difficulties when designing anti-phishing software. We therefore conducted research to discover how users interpret phishing emails.

We analyzed a number of phishing/spam samples from invited 6 participants from different countries and extracted useful characteristics from these emails to classify the emails. These 6 participants are all acquaintances of the authors of the papers. They are from four different countries, i.e. Finland, United Kingdom, Greece and China. We asked the participants to forward to us each email they received and they thought was spam or phishing email. We also designed and implemented a survey to disclose how the participants felt that phishing/spam emails disturbed them. We gathered answers from the survey with a set of pre-defined criteria.

From our investigation on reported phishing/spam samples, we found that phishing/spam emails were not always distinct from legitimate ones. In addition, friends' email addresses were used to easily convince victims to believe the authenticity of the emails. Language barriers also prevented successful phishing. Because of a lack of an opt-out option in some e-newsletters, participants classified these subscriptions as phishing/spam emails.

From the survey, we also collected valuable feedback. Many interesting responses were given. For example, participants replied that spam messages are time-consuming and an annoying part of their daily lives. In addition, we found that different people may apply different interests and different criteria to define spam/phishing emails. Participants suggested the use of a separate email address for work-based emails.

Although the number of email samples and participants in this survey was limited, we found that among different phishing attacks, users have different feelings and requirements regarding the anti-phishing software. Further analysis of the results and survey feedback found that a drastic and influential approach towards the protection of email users needs to emerge that combines three important issues: (i) software user psychology, (ii) human-centered software design quality criteria and (iii) the cognitive profiles of software/email exploiters. In this way, future information systems may be designed with generally verified and acceptable quality criteria as well as enhanced security and preventive maintenance, and they would also be required to take into account software users' psychology and their usability needs.

4.2.2 Paper 4 – Heuristic Evaluation of Anti-phishing Toolbars

Heuristic evaluation enables the quick detection of usability problems in anti-phishing toolbars. However, Nielsen's heuristics are not designed to evaluate specific software. Our heuristics were therefore specialized for anti-phishing software (Li and Helenius 2007). For anti-phishing software, the 13 heuristics (listed below) are similar but with greater consideration of the context of phishing and its prevention. These differences were designed deliberately. This is because anti-phishing toolbars are a form of security- and usability-critical software. For usability experts in our heuristic evaluation, we had to introduce the heuristics from the usability perspective. The anti-phishing software heuristics are explained as follows:

1. *Visibility of system status.* This heuristic inspects the visual capability of the toolbars. Visual capability should be checked in three stages: visibility before checking the authenticity of the website, visibility during checking and visibility of the result. In each stage, anti-phishing toolbars should always keep users aware of what is going on and of the result of identifying the web page. Moreover, response times and types should be reasonable and appropriate.
2. *Match between system and the real world.* Most vulnerable users do not have sufficient knowledge of computers and the Internet. From this heuristic, each operation of the anti-phishing toolbar should be understandable and predictable for non-sophisticated users. This means that people who have no professional knowledge of computers or e-commerce should still be able to protect themselves based on instructions or warnings from toolbars.

3. *User control and freedom.* As mentioned in the second heuristic rule, we should not expect e-commerce customers to have to learn about computers in depth before using them. Anti-phishing toolbar designers should not assume that every user is able to operate each functionality of the toolbar correctly or as expected. Furthermore, it is necessary to provide additional functionality for users to undo and redo their actions when they realize that they are incorrect. In addition, it should be possible for users to leave an unwanted situation before completing a transaction.
4. *Consistency and standards.* This requirement originates from the system requirements. For example, it is difficult to encourage a Microsoft Windows user to become used to another system unless its user interface resembles that of Microsoft Windows. The same is true for anti-phishing toolbars. The language used in toolbars should follow similar platform and browser conventions. Moreover, advice should be consistent when the same risk level of suspicious web pages or emails is detected.
5. *Help users to recognize, diagnose and recover from errors.* When users successfully pass a validation and carry out problematic or incorrect procedures, the toolbars should alert or provide advice to help the users to correct and recover from errors. This advice should be offered before, during and after users make decisions.
6. *Error prevention.* Similar to the third heuristic rule, error prevention can also avoid crashes or other potential problems caused by incorrect user operation. Toolbars should provide necessary checks or confirmations before any action is committed. Unlike the third heuristic rule, error prevention focuses on validating each operation and input by users, instead of undo functionality.
7. *Recognition rather than recall.* Any user, whether sophisticated or not, should be able to make a correct decision that prevents phishing without having to carry out complicated sequences of operation. Every warning or recommendation provided by a toolbar should be sufficiently understandable. In this way, users do not need to worry about being compromised due to forgetting the correct instructions, even if the users operate the toolbar incorrectly.
8. *Flexibility and efficiency of use.* In order to prevent phishing, typically users have to make some action when a fraud attempt is detected. However,

sometimes expert users are familiar with how to prevent specific phishing attempts when a warning arises, and they do not want to read repeated explanations. In this case, flexibility and efficiency of the toolbar should facilitate operations by experienced users and enable them to skip repeated instructions or carry out a pre-saved default operation. Obviously, flexibility may also cause mis-operation or new vulnerabilities. Therefore, users should also be able to return to the default settings.

9. *Aesthetic and minimalist design.* This heuristic mainly concentrates on the concision of the user interface of toolbars. The task of anti-phishing toolbars is to assist users to identify and stop fraud, not, for example, commercial promotion. It is therefore more meaningful and important to make sure that only phishing-prevention-related information is present in the toolbars. Concise and well-designed toolbars will not confuse users about what they should take into account and what to do next when a warning is displayed.
10. *Help and documentation.* Users are not omnipotent, and they need to learn how to use different anti-phishing toolbars by themselves. In this case, user manuals, tutorials and instant help should be available with the toolbars.
11. *Skills.* Phishers usually take advantage of users' lack of knowledge about networks and operating systems (Dhamija et al. 2006). Therefore, toolbars should support, extend, supplement and enhance users' skills and background knowledge of phishing prevention. Enhancements should be made from the client side only, because it is not useful to evaluate the capabilities of toolbars against all types of phishing techniques.
12. *Pleasurable and respectful interaction with the user.* In this heuristic evaluation rule, we attempted to determine the level of convenience of anti-phishing toolbars for users. Both functionality and aesthetics should be considered.
13. *Privacy.* Toolbars are used for protecting users' confidential information from being abused or stolen. However, some toolbars also need to know personal information about users, such as contact methods. This type of information should also be carefully protected when toolbar providers obtain such information.

When the heuristics were ready, evaluators, who were selected from usability experts in the University of Tampere, checked against these heuristics. They determined the usability issues and rated the severity of these issues on a scale from major, severe and minor to cosmetic. By analyzing the results of this heuristics evaluation, anti-phishing toolbar designers may be able to improve their design as follows:

1. *Main user interface of the toolbar*. According to our findings, the main user interface of the toolbar is very important. First, the status of the toolbar should be shown appropriately. This means that when browsing a web page, a user should be able to observe easily what the toolbar is doing and whether or not the current web page is authentic. Second, the anti-phishing client-side application interface should be sufficiently simple so that it is easy to understand and it does not occupy too much space in the browser's interface. Of course, frequently used and important functionalities should be sufficiently easy to find, such as configuration settings, viewing the results of website identity analysis and reporting a suspicious web page or a web page wrongly evaluated as suspicious. Some parts of the interface design of SpoofGuard are a good example of this, such as its traffic light indicator and Options button, which informs the user about available and frequently used functionalities.
2. *Warnings*. Because of a lack of a reliable strategy to detect fraud, application warnings need to be carefully designed. It is important that a user is able to react correctly when a fraudulent or suspicious web page is found. According to the evaluation by Zhang et al. (2007) and their observations, false and undetermined detection is not a minor issue. It would be problematic if a user relies only on toolbars with fixed detection algorithms. Therefore, there should be at least four levels of security indication: warning of detected web forgery, warning of a web page determined to be suspicious, a warning that a web page is unable to be determined, and an indication of an authentic web page. The warning given by Google Safe Browsing is a good example of highlighting web forgery. Google's warning may prevent incorrect visits by users. The warning for a suspicious page may be the same as that for a forgery. The differences between them may be in the given recommendations and their indications. For example, there may be only one recommendation

(stop visiting) available for a forged web page, and the indicator for the warning may be a stop sign. However, there may be two further recommendations (stop visiting, or check authenticity manually) when a suspicious page is found. Likewise, the indicator for a suspicious page should not be as strong as that for a forgery (e.g., an exclamation mark). An undetermined web page should also be notified to users. When this type of page is found, instant help documentation or a manual is needed to help the user identify the page manually. Additionally, cleared pages should be indicated in a similar way for consistency. For instance, the indicator could be shown at the same location as for other levels of phishing warning indicators. In their usability study, Lorentin and Karvonen (2008) conducted a detailed usability evaluation of anti-phishing toolbar indicators, and a user-preferred security indicator was suggested. Finally, a double warning should be used if an erroneous choice is made. For example, if a user accidentally makes a decision that leads to them visiting a phishing website, a second warning should be made to enable the user to correct the mistake.

3. *Help system*. Compared to other software, a client-side anti-phishing application should help users on some occasions. These occasions may include when users make a dangerous choice, when they are confused by some terms and when they want to learn how to identify the correct service manually. To maximize the efficiency and convenience of such help, different ways of showing the help may be used for different occasions. For example, when a user attempts to find further advice, an instant help system is needed. Another example is when a user needs to determine the consequences of different choices when a warning is presented. However, text that may help the user to understand certain terms or the consequences of certain choices must not be included with the warning, because too much information will confuse users. On other occasions, online help documentation may be better, because more information can be provided. A good example is the help system of the Netcraft toolbar.

Our findings may be helpful for designers to improve anti-phishing software, but there are limitations: the lack of end-user feedback, the limited number of software designs tested.

4.2.3 Paper 5 – Towards a contingency approach with whitelist- and blacklist-based anti-phishing applications: what do usability tests indicate?

Many different anti-phishing applications were designed and developed (Li et al. 2007), and towards this contingency approach we need to research on the usability issues. Although our heuristic evaluation determined methods of improving anti-phishing toolbars, user feedback and behavior concerning anti-phishing toolbars was not collected. Therefore, we designed and conducted a usability test on anti-phishing toolbars. According to our preliminary analysis, existing anti-phishing toolbars are based on either blacklist or whitelist detection mechanisms. In addition, we implemented our own whitelist-based anti-phishing toolbars, IEPlug. In order to compare these two types of toolbars from the perspective of usability, we tested Google Safe Browsing (blacklist-based) and IEPlug (whitelist-based) toolbars. We invited 20 student participants from the University of Tampere. None of them had sufficient security knowledge about e-commerce, as was our intention.

The test procedure consisted of four sessions: tutorial session, interview session and two test sessions (Figure 9). In the tutorial session, instructions on how to use the tested toolbar were given according to those prepared by the software provider. In the first test session participants tested the anti-phishing software according to their understanding of it. In the interview session, the participants were given face-to-face tuition on how the anti-phishing toolbar being tested works. The second test session involved testing the same anti-phishing software following users' experiences and further introductions from researchers in the interview session. The participants were asked to state the authenticity of given test web pages, which were stored at a server connected in a secure environment. During the test, all the participants were asked to think aloud so that researchers could monitor their mental activities.

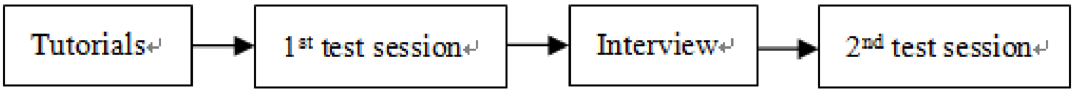


Figure 9. The usability test procedure

After the usability test, the recorded user behaviors were analyzed. No significant difference was found in the accuracy rate measuring how well participants identified the tested web pages with the two toolbars. However, statistical data from both toolbars showed that the accuracy rate increased after the interview session. Moreover, other key findings from the usability test were found to be potentially useful in helping to improve the quality of anti-phishing software:

1. Understanding the Domain Name. Seven participants for Google Safe Browsing and seven participants for IEPlug checked the domain name during the test. Although all participants successfully extracted the domain name before the test, only three participants used the domain name to identify the web page during the first test session.
2. SSL Awareness. Security certificates for web servers, as a form of web page identity, were felt by participants to be more technical than checking the domain name. In fact, such certificates were too complex for most participants. Sixteen out of twenty participants had no sufficient knowledge about SSL or TSL and thus did not know how to verify the certificates.
3. Security Hints on Web Pages. Security hints or instructions on web pages are very helpful in aiding users to decide whether to trust a service. However, our usability tests revealed that these hints have weaknesses. Firstly, it is easy to abuse these hints. The hints are usually in the form of text or a link on the web page, and phishers can easily change them. Secondly, some of these hints are ambiguous. For example, one was simply to "click the lock," which misled one of our participants to click the lock icon on the web page.
4. Regular Elements on Web Pages. Another valuable finding was that during the first test session seven Google Safe Browsing participants and four IEPlug participants preferred checking layouts or some insignificant functions on web pages. For example, the most used method was to try the links on the page. In the first test session, seven Google Safe Browser and five IEPlug participants clicked at least one link in order to check the authenticity of the website. Favored links were "help" or "privacy policies," where participants expected to find information to help them verify the authenticity of the website. This finding also demonstrates the relationship between web page aesthetics and users' trust behavior, as observed by Karvonen (2000).

5. Spoof Awareness. Interestingly, over the entire test procedure three participants could not fully understand the definition of an authentic web page. They appeared to believe that any web page that asks for a username and password is an authentic web page.

Although the findings from the usability test were interesting and useful in improving anti-phishing toolbars, there were also some limitations. For example, unless a large amount of feedback is collected, analysis remains a problem, and it is difficult to draw general and useful conclusions. To simplify this problem, user behavior in a phishing context was studied and modeled with finite-state machine methodology.

4.2.4 Paper 6 – Overview of User-Centered Quality Assurance Methodologies for Anti-phishing Software and Phishing-Resistant Systems

I summarized all user-centered quality assurance methodologies used for anti-phishing software and phishing-resistant systems. This paper gathered the important findings from our previous related research and critiqued their user-centered quality assurance methodologies.

In our previous research, four quality assurance methodologies were applied: misuse case methodology, end-user survey, usability research methodologies and user-behavior modeling methodology. The first three methodologies were thoroughly discussed in Papers 1, 3, 4 and 5). From these previous studies, I was able to collect the requirements and potential benefits for end users, anti-phishing designers and developers. For example, end users may be able to further understand potential vulnerabilities when they use online applications, e.g., how to reliably identify web pages and how to check their authenticity. I also critiqued the methodologies in our anti-phishing research. For anti-phishing designers and developers, these methodologies could be applied in their future work, or they may even improve these methodologies in order to eradicate their drawbacks, for example, by investigating the heuristics used to evaluate the design of anti-phishing software. Although I collected some anti-phishing requirements, it was difficult and costly to manage these requirements. Information system modeling is common and useful, especially for the design of complicated systems (e.g. Veijalainen and Weske 2002, Veijalainen 2007). To lower the maintenance costs, I also plan on designing a study to extract a user-behavior model in a phishing context with the formal finite-state machine modeling methodology.

The limitation of this research is that not many methodologies were introduced, even though every methodology involved is well-selected. In this ongoing research, I first intend to establish a typical example of phishing. By observing user behavior in a phishing context and previous research findings on decision-making theories, I aim to design a user-behavior model with a finite-state machine in a phishing context (Figure 10). The states of the finite-state machine are collected and abstracted from our observations during our previous surveys and experiments. The inputs are a sequence of decisions a user makes in the phishing context. Besides finite-state machine theories, I will use the Labeled Transition System Analyzer (Magee et al. 2010) to facilitate the design of our model and verification tasks.

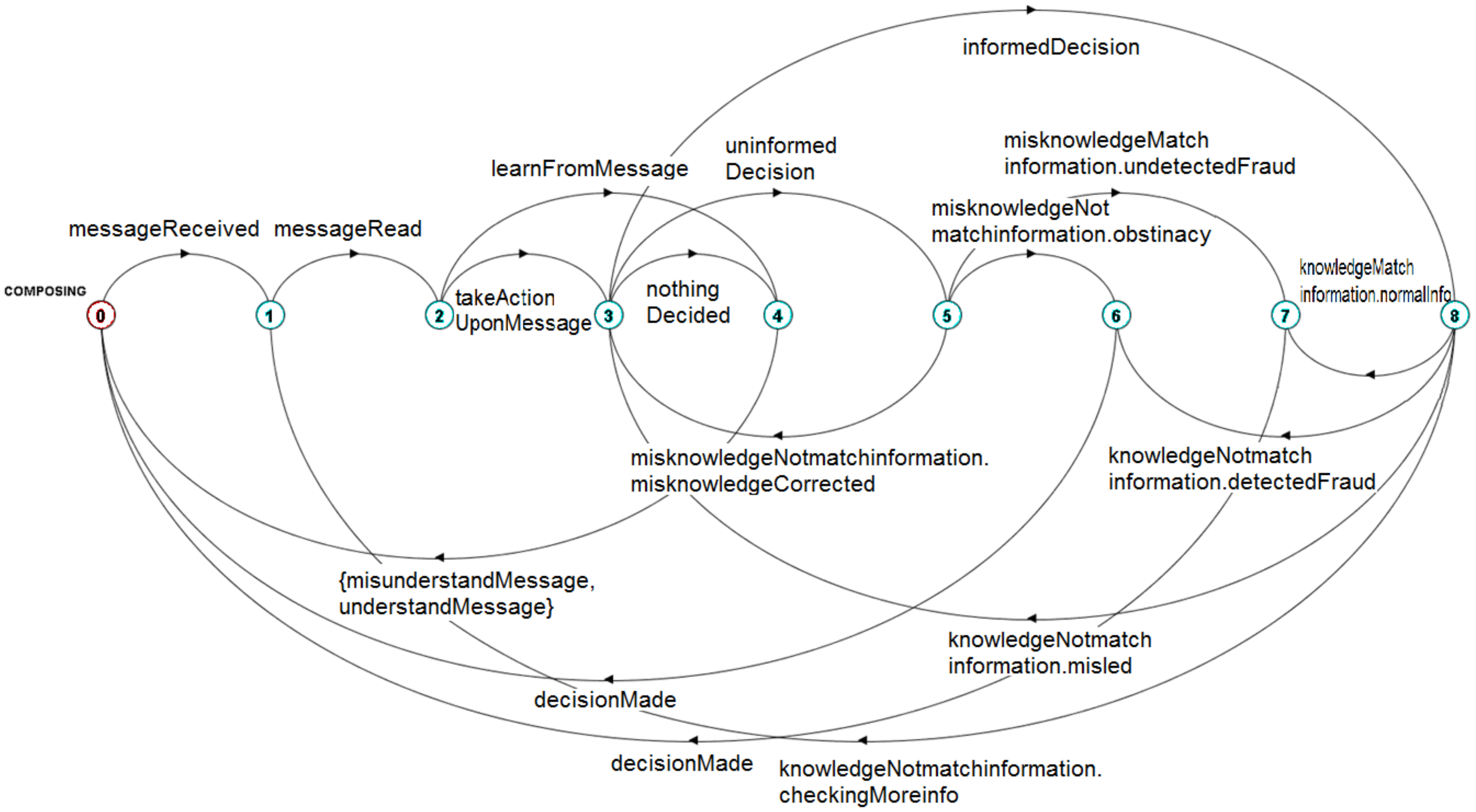


Figure 10. Finite-state machine diagram generated by the Labeled Transition System Analyzer script of user behavior in a phishing context, the texts above/below the arrows are the inputs of the system.

4.2.5 Paper 7 – New Usability Metrics for Authentication Mechanisms

Because of the security and usability vulnerabilities in authentication mechanisms, phishers are able to successfully conduct phishing scams. Furthermore, these scams are becoming increasingly advanced. In order to aid the design of usable and secure authentication mechanisms, a set of software quality metrics are needed. We deduced this new set of usability metrics from existing representative authentication mechanisms and online identity threats.

First, we analyzed the characteristics of five representative online systems based on three widely known authentication factors: *what you know*, *what you have* and *what you are*. Our analysis highlighted that the secrets for authentication must be easily and effectively remembered, used, reused and maintained with sufficient confidentiality. This showed that both security and usability must be considered when designing a new set of usability metrics as a guideline for authentication mechanisms.

According to the findings of the analysis, we proposed nine usability metrics:

1. Customizability
2. Learnability
3. Maintenance of credentials for multiple online services
4. Efficiency
5. Quality of help system
6. Replicability
7. Effectiveness
8. Non-intrusiveness
9. Cost-effectiveness

Customizability means that users are able to select their own credentials for authentication. *Learnability* means that users are able to easily learn how to use the authentication mechanisms. *Maintenance of credentials for multiple online services* concerns the amount of effort that end users need to make to maintain their credentials across multiple online services such as emails, online banking, online payment services and social networking. *Efficiency* examines how much time and effort is expended when users are authenticated. *Quality of help system* refers to how easily users are assisted when using authentication methods. The *replicability* metric describes whether the information involved in authentication mechanisms can be replicated. If the information is hard to be replicated, it can be hard for phishers to make a phishing web page which looks like an authentic one. *Effectiveness* describes how effectively authentication mechanisms and users are able to identify each other. *Non-intrusiveness* verifies the integrity of the whole authentication process. *Cost-effectiveness* evaluates whether the extra costs of authentication mechanisms lower the likelihood of financial losses and offer more pleasant user experiences. Compared with ISO-9241-11(2010), our criteria concentrate on the usability of authentication methods in the current situation of phishing evolution and online services. In this set of quality criteria, more usability metrics were added. Firstly, *maintenance of credentials for multiple online services* relates to the situation where a user has had different online services. *Quality of help system* is a new metric because phishing may be successful if the victims' knowledge is not adequate to identify the phishing information (see Figure 10). *Non-intrusiveness* helps the anti-phishing software designers to pay attention to preventing man-in-the-middle and XSS phishing attacks (Asokan et al. 2003, Evers 2007). *Cost-effectiveness* is to also consider costs when nowadays web service providers use different security devices to prevent phishing (e.g. Figure 8).

After introducing this set of usability metrics, we examined existing authentication methods. In our evaluation, we found that none of the methods performed perfectly. We concluded that the above set of usability metrics was able to help strengthen service quality and may act as a design guide to aid the creation of usable and secure online authentications. In future research, these mechanisms would be further tested practically, and we would make further research efforts to determine weighted values for the metrics in order to draw conclusions on the trade-offs of security and usability for authentication mechanisms.

The collection of the research questions and publications in my thesis research can be found in Table 6.

Table 6. Thesis research questions and publications

Research questions	**Publications**
How can the quality of	**Paper 1** –Li L., Helenius M., Berki E. (2007).

anti-phishing software and phishing-resistant system be improved with engineering methodologies?	Phishing-Resistant Information Systems: Security Handling with Misuse Cases Design, *Proceedings of Berki, E., Nummenmaa, J., Sunley, I., Ross, M. & Staples, G. (Eds) Software Quality in the Knowledge Society*, SQM 2007. Tampere, Finland,1-2 August 2007, pp. 389-404. **Paper 2** –Li L., Berki E., Helenius M. (2011). Evaluating the Design and the Reliability of Spam/Phishing Content Filtering Performance Experiments, *Proceedings of Dawson, R., Ross, M., Staples, G. (Eds), Global Quality Issues, SQM 2011*, Leicestershire UK, 18 April 2011, pp.339–357.
How can the quality of anti-phishing software and phishing-resistant systems be improved with end-user studies?	**Paper 3** – Li L., Helenius M., Berki E. (2011). How and Why Phishing and Spam Messages Disturb Us? *Proceedings of Bradley G. (Ed) IADIS International Conference ICT, Society and Human Beings 2011*, Rome, 20-26 July, 2011, pp.239–244. **Paper 4** – Li L., Helenius M. (2007). Usability Evaluation of Anti-phishing Toolbars, Journal of Computer Virology 2007 (3), pp.163–184. **Paper 5** – Li L., Berki E., Helenius M., Ovaska S., (2012). Towards a contingency approach with whitelist- and blacklist-based anti-phishing applications: what do usability tests indicate? Submitted to: Behaviour & Information Technology Journal, (accepted, to be published). **Paper 6** – Li L. (2012). Overview of User-centered Quality Assurance Methodologies for Anti-phishing Software and Phishing-resistant Systems, *Proceedings of Berki, E., Valtanen, J., Nykänen P., Ross, M., Staples, G., Systä K. (Eds), Quality Matters*, SQM 2012, Tampere, Finland, 20-23 August 2012, pp. 11-20.

	Paper 7 –Li L., Berki E., Helenius M., Savola R. (2012). New Usability Metrics for Authentication Mechanisms, *Proceedings of Berki, E., Valtanen, J., Nykänen P., Ross, M., Staples, G., Systä K. (Eds), Quality Matters*, SQM 2012, Tampere, Finland, 20-23 August 2012, pp. 239-250.

5. Conclusions and Future Work

Phishing attacks combine social engineering and information technologies and not only take advantage of the cognitive vulnerabilities of users but from a security perspective also exploit design flaws in information systems. The nature of phishing means that it is difficult to detect and prevent. Therefore, various anti-phishing solutions (SpoofGuard 2001, Netcraft 2005, Google Safe Browsing 2006, DOMAntiPhish (Rosiello et al. 2007), Anti-phishing IEPlug 2007) and phishing-resistant authentication designs (e.g., electronic tokens, one-time password) have been implemented and released. However, the quality of these phishing prevention systems has so far not been studied. To facilitate their quality assurance and management, I used a contingency approach and studied the research area in terms of the user-centered quality assurance of anti-phishing applications and phishing-resistant systems.

As one contribution of this thesis, a contingency and holistic research framework was proposed. This research framework consists of several user-centered quality assurance research methods, including software-engineering methodologies, heuristic evaluation and usability experiments. Similar to Multiview, my research framework combines multifaceted aspects of anti-phishing activities, particularly on the methods of investigating techno-social issues in the phishing context. Social- and human-related research was investigated with end-user surveys and usability experiments. Technical aspects in phishing-resistant systems can be investigated with software quality assurance methodologies and theories. Using these different research methods in the anti-phishing contingency framework, designers and developers of phishing-resistant systems should be able to effectively collect phishing-related requirements in order to guarantee the quality of the targeted systems.

In addition to creating this research framework, many valuable research findings were collected. These research findings consist of two parts. The first part involves evaluating and assuring the quality of anti-phishing and phishing-resistant software with misuse cases at the design stage and the performance evaluation methodology and metrics for text-content-filtering algorithms. The second part shifted the research focus to end-user research methodologies, including heuristic evaluation, end-user surveys, usability testing and user-behavior modeling.

In the first part, the research on misuse-case-based phishing prevention design showed that misuse cases are able to detect vulnerabilities to phishing, and designers are able to extract anti-phishing-related security requirements with this methodology at the design stage. Although this method may help to improve the quality of anti-phishing software and phishing-resistant systems at the design stage, its limitations cannot be ignored, for example the extra maintenance costs in the development process and the uncertain quality of designed misuse cases. A literature review of tests on the performance of text-content-filtering algorithms was conducted to collect performance metrics on phishing content filtering. After the literature review, the challenges and limitations of existing performance evaluation tests were analyzed. For future performance evaluations, suggestions were made for designing a reliable performance test.

In the second part, the research mainly focused on using end-user investigations to improve the quality of phishing prevention and phishing-resistant systems. Multiple participants were invited to carry out a variety of investigations. In the survey on how users were disturbed by phishing/spam messages, six participants from four countries were invited to submit phishing/spam messages classified from their email inboxes based on their own understanding of what constitutes phishing/spam messages. Although the number of participants was limited, by analyzing reported message samples, designers of anti-phishing solutions should be able to determine how end users interpret phishing messages and evaluate current phishing filters of email services. Participants were also asked to complete a survey to describe how phishing/spam messages disturb their daily lives and what they expected from anti-phishing features. With the help of these findings, designers should be able to have more user requirements for designing the next generation of anti-phishing features for email services.

I also evaluated the existing anti-phishing software and collected what other requirements are hidden when the anti-phishing software is in use. One heuristic evaluation and one usability test were also carried out to determine the usability problems of existing anti-phishing toolbars and to collect feedback from end users on their understanding of phishing web pages and two anti-phishing toolbars (whitelist- and blacklist-based toolbars). These findings should help designers of anti-phishing solutions to upgrade their current solutions from the perspective of usability. Although valuable feedback from end users was collected, the feedback varied for different individuals.

To conclude the study, an overview was made of user-centered quality assurance methodologies, and a set of user-centered quality assurance criteria for anti-phishing and phishing-resistant systems was proposed. In Paper 6, I reviewed all of the four user-centered quality assurance methodologies and their findings. In addition to critiquing the methodologies in Papers 1, 3, 4 and 5, I presented a user-behavior modeling methodology in a phishing context. According to the research, there are two types of user behaviors in the phishing context: acquiring external information and making decisions with internal information acquired from external sources. To prevent users making critical erroneous decisions, anti-phishing applications and phishing-resistant systems should not focus solely on scenarios where users must make decisions to take actions, such as login mechanisms to authenticate users, but should also take into account all of the information presented to end users. To assist users in making correct decisions, users' decision-related actions should be monitored carefully. For example, in Paper 4, it was found that an appropriate warning can prevent dangerous behavior and helps the user make decisions when the user visits a phishing web page. In Paper 5, it was found that people attempted to look for assistance in the contents of web pages. In Paper 3, it was found that some users felt that it was safer to have separate work-based email addresses and personalized detection. Another conclusive finding was presented in Paper 7. In that paper, I proposed a new set of usability metrics for authentication mechanisms. From the research, it was found that in designing a usable and secure authentication system it is essential to equip it with quality characteristics such as customizability, learnability, ease of maintenance, efficiency, high-quality help system, uniqueness, non-intrusiveness and cost-effectiveness. These measurable metrics may also be useful in the future quality assurance of different phishing-resistant systems.

It is good to see that many open-source efforts (e.g., SpamAssassin 2011, DomainKeys Identified Mail 2011) have contributed to anti-phishing applications and phishing-resistant frameworks. However, comparisons between and debates over open-source and proprietary software continue. Some believe that open-source software is more secure than the closed-source equivalent. This viewpoint is supported not only by the comparisons conducted by Samoladas and Stamelos (2006) but also by software-engineering researchers (Wheeler 2003). Wheeler concluded that a closed source cannot guarantee the security of programs. Instead, with more reviews, open discussions and tests by the public, open-source software performs better in terms of security. When a vulnerability is found, it can be patched immediately. However, with closed-source software, vulnerabilities cannot be patched as quickly unless these vulnerabilities are exposed to the public and attract a large amount of attention.

It is also essential to examine the advancements taking place in computing. Today, as the network evolves towards ubiquitous computing, more and more computing devices are getting connected. These different computing devices and user interfaces are designed by different online services. This means that the entire system has to be operated by users who need to acquire sufficient knowledge about how to identify authentic online services and how to securely use their identities online. Some researchers believe that, when certain user knowledge is absent in a phishing context, it is possible for the user to be spoofed when making decisions (Dong et al. 2008, Kumaraguru et al. 2006). In that case, the interoperations among different devices may increase the potential for successful phishing scams, because the scope of the knowledge needed to manage the entire ubiquitous computing system is tremendously extended. Therefore, there is a need for a shared, secure and standardized design for phishing-resistant systems, which can act as a platform (e.g. Kostiainen and Asokan 2011) for common Application Programming Interfaces and security-enhancement features. In addition, the current research is about software quality engineering, but the future work can be combined with research advancements from other fields such as artificial intelligence, content filtering, and anti-phishing software for social media.

The findings and conclusions of my research may be applied in the future development of phishing prevention systems and software quality management. These findings should help developers and designers of phishing prevention systems to improve their understanding of what end users expect from these prevention strategies, by revealing how and why spam/phishing messages disturb end users (time consumption, danger etc.), and help the designers to more effectively and efficiently protect different end users, by establishing (i) three key components of anti-phishing software (main user interface, warnings, help system); (ii) how users identify authentic web services (identities are mainly obtained from web page contents); (iii) how to improve the quality of help systems or tutorials provided with anti-phishing software to aid users in identifying phishing pages; (iv) how to prepare corpora with different topics, in different languages and with non-skewed distributions; and (v) usability metrics as a guide to help in the design of new authentication methods.

References

Androutsopoulos I., Paliouras G., Karkaletsis V., Sakkis G., Spyropoulos C. D., Stamatopoulos P. (2000a). Learning to Filter Spam E-Mail: A comparison of a naive Bayesian and a memory-based approach. Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000), 1-13, Lyon, France

Androutsopoulos I., Koutsias J., Chandrinos K.V., Spyropoulos C.D. (2000b). "An Experimental Comparison of Naive Bayesian and Keyword-based Anti-spam Filtering with Encrypted Personal E-mail Messages. Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000), 160-167, Athens, Greece

Anti-phishing IEPlug (2007). www.cs.uta.fi/~ll79452/ap.htm, retrieved on 2nd May 2007

APWG. (2012). Anti-phishing Working Group Trend Reports, http://www.apwg.org/reports/apwg_trends_report_q1_2012.pdf, retrieved on 11th Oct., 2012

Askola K., Puuperä R., Pietikäinen P., Eronen J., Laakso M., Halunen K., Röning J. (2008). Vulnerability Dependencies in Antivirus Software. Second International Conference on Emerging Security Information, Systems and Technologies (SECURWARE '08), Cap Esterel, France, pp. 273-278.

Asokan N., Niemi V., Nyberg K. (2003). Man-in-the-middle in tunnelled authentication protocols. Proceedings of the 11th international conference on Security Protocols, Bruce Christianson, Bruno Crispo, James A. Malcolm, and Michael Roe (Eds.). Springer-Verlag, Berlin, Heidelberg, pp28-41.

Avison D.E. (1990). A contingency framework for information systems development. PhD thesis, Aston University.

Avison D., Wood-Harper T. (2003). Bringing social and organisational issues into information systems development: the story of multiview. In Socio-technical and human cognition elements of information systems, Steve Clarke, Elayne Coakes, Gordon M. Hunter, and Andrew Wenn (Eds.). IGI Publishing, Hershey, PA, U.S.A. pp5-21.

Balfanz D., Durfee G., Smetters D.K. (2005). Making the Impossible Easy: Usable PKI, Security and Usability, O' REILLY publishing house, pp319-329

Barth A. (2011). HTTP State Management Mechanism, RFC 6265, IETF

Bellamy-McIntyre J., Luterroth C., Weber G. (2011). OpenID and the Enterprise: A Model-Based Analysis of Single Sign-On Authentication, Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International

Berki E. (2006). Examining the Quality of Evaluation Frameworks and Metamodeling Paradigms of Information Systems Development Methodologies. Book Chapter. Duggan, E. & Reichgelt, H. (Eds) Measuring Information Systems Delivery Quality. Pp. 265-289, Idea Group Publishing: Hershey, PA, USA.

Berki E., Georgiadou E. (1996). Towards resolving Data Flow Diagramming Deficiencies by using Finite State Machines. I M Marshall, W B Samson, D G Edgar-Nevill (Eds). Proceedings of the 5th International Software Quality Conference. Universities of Abertay Dundee & Humberside, Dundee, Scotland, Jul 1996.

Berki E., Georgiadou E., Holcombe M. (2004). Requirements Engineering and Process Modelling in Software Quality Management – Towards a Generic Process Metamodel. The Software Quality Journal, 12, pp. 265-283. Kluwer Academic Publishers

Berki E., Jäkälä M. (2009). Cyber-Identities and Social Life in Cyberspace. Hatzipanagos, S. & Warburton, S. (Eds) Social Software and Developing Community Ontologies (London: Information Science Reference, an imprint of IGI Global). Pp.28-40.

Brown M., Housley R. (2010). Transport Layer Security (TLS) Authorization Extensions, RFC 5878, IETF, http://tools.ietf.org/html/rfc5878, retrieved on 16th Jan. 2013.

Chhabra S., Aggarwal A., Benevenuto F., Kumaraguru P. (2011). Phi.sh/$oCiaL: the phishing landscape through short URLs. In Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS '11). ACM, New York, NY, USA, pp.92-101.

Chen X. (2008). Guangdong Police Lists 12 Types of Phishing Scams Deployed through Mobile Phones, http://news.sina.com.cn/c/l/2008-07-08/104815893095.shtml, retrieved on 11th Dec., 2011

Christey S. (2011). CWE/SANS Top 25 Most Dangerous Software Errors, http://cwe.mitre.org/top25/archive/2011/2011_cwe_sans_top25.pdf, retrieved on 29th Dec., 2011

Cormac G., Kolcz A. (2009). Spam Filter Evaluation with Imprecise Ground Truth. Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, 604-611, Boston, Massachusetts, USA

Cormack G. V., Lynam T. R. (2007). Online Supervised Spam Filter Evaluation, ACM Transactions on Information Systems, 25(3)

Commtouch (2011). Recurrent pattern Detection Technology White Paper. http://www.commtouch.com/download/367, retrieved on 12th Nov., 2011

Coventry L. (2005). Usable biometrics, Security and Usability, 2005, O’ REILLY publishing house, pp175-195

Cranor F. L., Garfinkel S. (2004). Guest Editor’s Introduction: Secure or Usable? 2(5), IEEE Security & Privacy, September/October 2004, pp. 16-18.

Daniels A., Yeates D. A. (1971). *Basic Training in Systems Analysis*.London: Pitman.

Dhamija R., Tygar J.D., Hearst M. (2006). Why phishing works. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp581-590.

Dhillon G., Moores T. (2001). Internet privacy: Interpreting key issues. *Information Resources Management Journal*, Oct 2001, Issue (14:4), p 33-37.

Dierks T., Rescorla E. (2008). The Transport Layer Security (TLS) Protocol Version 1.2, RFC 5246, IETF.

Dinev T. (2006). Why spoofing is serious internet fraud, Communication of the ACM, 49(10)p76-82.

DNSSEC (2011). http://www.dnssec.net/, retrieved on 10th Sep., 2011

Dong X., Clark J. A., Jacob J. (2008). Modelling user-phishing interaction. Proceedings of Human-System Interaction, May 25--27, 2008, Kraków, Poland.

Donnerhacke L. (2011). Securing BGP, invited talks at INFORTE seminar, Tampere, Finland, on 15th September, 2011

Duquenoy P., Thimbleby H., Torrance S. (1999). Towards a synthesis of Discourse Ethics and Internet regulation. Proceedings of ETHICOMP 99, LUISS Guido Carli,Centro di Ricerca sui Sistemi Informativi, Roma, 1999.

Duquenoy P. (2007). "Ethics in the environment of the Information Society" invited presentation at the European regional Conference on the "ethical dimensions of the information society: Ethics and human rights in the information society" organised by the French Commission for UNESCO in cooperation with UNESCO and the Council of Europe, 13-14th September, 2007, Strasbourg.

Egelman S., Cranor F. L., Hong J. (2008). You've Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings, proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, p1065-1074.

Episkopou D. M., Wood-Harper A. T. (1985). The Multiview methodology:Applications and implications. In Bemelmans, T. M. A. (Ed.), BeyondProductivity:Information Systems Development for Organisational Ef-fectiveness. Amsterdam: North Holland.

Eronen J., Karjalainen K., Puuperä R., Kuusela E., Halunen K., Laakso M., Röning J. (2009). Software vulnerability vs. critical infrastructure - a case study of antivirus software. International Journal on Advances in Security, 2(1), pp.72 - 89.

Evers J. (2007). New tool enables sophisticated phishing scams, http://news.cnet.com/2100-1029_3-6149090.html, retrieved on 4th Nov. 2012.

F-secure (2012). F-secure Anti-virus Server Solutions, http://www.f-secure.com/en/web/business_global/products/servers/solution, retrieved on 31st Jan., 2012

Fawcett T. (2003). "In vivo" spam filtering: A challenge problem for data mining, KDD Explorations, 5(2) http://home.comcast.net/~tom.fawcett/public_html/papers/spam-KDDexp.pdf, retrieved on 12th Dec. 2010

Fiedler. E. F. (1963). A contingency model of leadership effectiveness, Urbana: Group Effectiveness Research Laboratory, University of Illinois.

Garfinkel S. (1995). Risks of Social Security Numbers, Communications of the ACM, October 1995. p. 146

Garfinkel S. (2003) Email-Based Identification and Authentication: An Alternative to PKI?, IEEE Security & Privacy, November/December 2003. pp. 20-26.

Georgiadou E., Siakas K. and Berki E. (2003). Quality Improvement through the Identification of Controllable and Uncontrollable Factors in Software Development. Messnarz R. and Jaritz K. (Eds) EuroSPI 2003: European Software Process Improvement, EuroSPI 2003 Proceedings, 10-12 Dec 2003, Graz, Austria. Pp. IX 31-45. Verlag der Technischen Universität: Graz.

Gmail Blog (2011). Advanced sign-in security for your Google account, http://googleblog.blogspot.com/2011/02/advanced-sign-in-security-for-your.html, retrieved on 12th Nov. 2011.

Google Safe Browsing (2006). http://www.mozilla.org/en-US/firefox/security/, retrieved on 3rd Dec., 2011

Haxdoor (2011). http://www.f-secure.com/v-descs/haxdoor.shtml, retrieved on 10th Sep., 2011

Helenius M. (2002). A System to Support the Analysis of Antivirus Products' Virus Detection Capabilities, *Ph.D Dissertation*, http://acta.uta.fi/pdf/951-44-5394-8.pdf, retrieved on 2nd Dec., 2011

Helenius M. (2006). Fighting against Phishing for On-Line Banking Recommendations and Solutions. Proceedings of the 15th Annual EICAR Conference Security in the Mobile and Networked World, Germany 2006, pp252-267.

Hong J. (2012). The state of phishing attacks. Commun. ACM 55(1), pp.74-81

Huber M., Mulazzani M., Leithner M., Schrittwieser S., Wondracek G., Weippl E. (2011). Social snapshots: digital forensics for online social networks. Proceedings of the 27th Annual Computer Security Applications Conference (ACSAC '11). ACM, New York, NY, USA, pp.113-122.

Hutchinson W., Warren M. (2000) Using the viable systems model to develop an understanding information system security threats to an organisation. Proceedings of the 1st Australian Information Security Management Workshop.

ISO-25010 (2011). http://nl.wikipedia.org/wiki/ISO_25010, retrieved on 12th Jan., 2012

ISO-9241-11 (2010). en.wikipedia.org/wiki/ISO_9241, retrieved on 13th Jan., 2011

Israr J., Guennoun M., Mouftah H. T. (2009). Mitigating IP Spoofing by Validating BGP Routes Updates, IJCSNS International Journal of Computer Science and Network Security, 9(5).

Jakobsson M., Ratkiewicz J. (2006). Designing Ethical Phishing Experiments: A study of (ROT13) rOnl auction query features. Proceedings of the 15th annual World Wide Web Conference, pp513-522.

Jakobsson M. (2006). Modeling and Preventing Phishing Attacks, Phishing Panel of Financial Cryptography

Jäkälä M., Berki E. (2004). Exploring the Principles of Individual and Group Identity in Virtual Communities. Commers, P., Isaias, P. & Baptista Nunes, M. (Eds) Proceedings of the 1st IADIS Conference on Web-based Communities. Lisbon. Pp 19-26. International Association for the Development of Information Society (IADIS): Lisbon

Järvinen P. (2010). On developing and evaluating of the literature review, Workshop on Literature Review in the IRIS31 Conference, Åre, Sweden, August 10-13, 2008. CD Publication.. www.cs.uta.fi/reports/dsarja/D-2008-10.pdf, retrieved on 25th Dec., 2010

Järvinen P. (2012). On Research Methods. OPINPAJAN KIRJA, Tampere, Finalnd, ISBN 978-952-99233-4-2

Kajava J., Anttila J., Varonen R., Savola R., Röning J. (2006). Information Security Standards and Global Business. Proceedings of International Conference on Industrial Technology (ICIT 2006), December 15-17, 2006, Mumbai, India, pp. 2091-2095.

Karvonen K. (2000) The Beauty of Simplicity. Proceedings of the ACM Conference on Universal Usability (CUU 2000), November 16-17, 2000, Washington DC, USA, pp. 85-90,

Karvonen K. Parkkinen J. (2001). Signs of Trust. Proceedings of the 9th International Conference on HCI (HCII2001), August 5-10, 2001, New Orleans, LA, USA

Kostiainen K., Asokan N. (2011). Credential life cycle management in open credential platforms (short paper). Proceedings of the sixth ACM workshop on Scalable trusted computing (STC '11). ACM, New York, NY, USA, pp.65-70.

Kumaraguru P., Acquisti A., Cranor L. (2006). Trust modeling for online transactions: A phishing scenario. Proceedings of Privacy Security Trust, Oct 30 - Nov 1, 2006, Ontario, Canada

Land F. (1998). A contingency based approach to requirements elicitation and systems development, Journal of Systems and Software, 40(1), pp.3-6.

Li L., Helenius M. (2007). Usability Evaluation of Anti-phishing Toolbars, Journal of Computer Virology 2007 (3), pp 163-184.

Li L., Helenius M., Berki E. (2007). Phishing-Resistant Information Systems: Security Handling with Misuse Cases Design, Proceedings of Software Quality in the Knowledge Society, Tampere, Finland, pp.389–404.

Li L., Helenius M., Berki E. (2011). How and Why Phishing and Spam Messages Disturb Us? Proceedings of IADIS International Conference ICT, Society and Human Beings 2011, Rome, pp.239–244.

Li L., Helenius M., Berki E. (2012). A Usability Test of Whitelist and Blacklist-based Anti-phishing Applications. Proceedings of MindTrek Academic Conference 2012, Oct 3-5, 2012, Tampere, Finland, pp195-202.

Lin W. T., Shao B. B. M. (2000). The relationship between user participation and system success: a simultaneous contingency approach, Information & Management, 37(6), pp.283-295

Litan A. (2004). Phishing attack victims likely targets for identity theft. FT-22-8873, Gartner Research

Lorentin B., Karvonen K. (2008). Enhancements to the Anti-phishing Browser Toolbar, Symposium On Usable Privacy and Security (SOUPS'08), Carnegie Mellon University, Pittsburgh, PA, U.S., New York, NY, USA 2008, ACM

Magee J., Kramer J., Chatley R., Uchitel S., Foster H. (2010). Labelled Transition System Analyzer, www.doc.ic.ac.uk/ltsa/, retrieved on 11th Dec., 2011

Mazhelis O., Markkula J., Veijalainen J. (2005). An integrated identity verification system for mobile terminals. Information Management & Computer Security, Emerald Group Publishing Limited, Vol. 13, Nr. 5 (Dec. 2005), pp. 367 – 378.

Microsoft (2011). User Account Control Step-by-Step Guide, http://technet.microsoft.com/en-us/library/cc709691%28WS.10%29.aspx, retrieved on 3rd Nov., 2011

Microsoft SenderID (2011). http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx, retrieved on 10th Sep., 2011

Microsoft SenderID Architecture (2011). http://www.microsoft.com/mscorp/safety/technologies/senderid/technology.mspx, retrieved on 19th Dec., 2011

NetCraft (2005). http://toolbar.netcraft.com/, retrieved on 30th Jan., 2006

Open DomainKeys Identified Mail (2011). http://www.opendkim.org/, retrieved on 11th Nov., 2011

Naumann J.D., Davis G.B., McKeen J.D. (1980). Determining information requirements: A contingency method for selection of a requirements assurance strategy. Journal of Systems and Software (1980), pp273-281.

Nielsen J. (1993). Usability Engineering, Morgan Kaufmann Publishing House.

Nikander P., Karvonen K. (2000). Users and Trust in Cyberspace, in: Christianson, Malcolm, Crispo and Roe (Eds.) Security Protocols, 8th International Workshop, Cambridge, UK, April 3-5, 2000; revised papers, LNCS 2133, Springer 2001 pp. 24-35,

Ngugi B., Kahn B. K., Tremaine M. (2011). Typing Biometrics: Impact of Human Learning on Performance Quality. Journal of Data and Information Quality (JDIQ) JDIQ 2(2), February 2011, Article No. 11.

Open Authorization (2011). http://oauth.net/, retrieved on 10th October, 2011

Piazzalunga U., Salvaneschi P., Coffetti P. (2005). Security and Usability, O' REILLY publishing house, pp221-242

Pressman R. S. (2001). Software Engineering – A Practitioner's Approach, Fifth Edition, McGraw-Hill.

Rootkit (2011), http://www.f-secure.com/en/web/labs_global/terminology-r#Rootkit, retrieved on 7th Jan., 2012

Rosiello A., Kirda E., Kruegel C., Ferrandi F. (2007). A Layout-Similarity-Based Approach for Detecting Phishing Pages. In IEEE International Conference on Security and Privacy in Communication Networks (SecureComm).

RSA FraudAction Research Labs, Phishing in Season: A Look at Online Fraud in 2012. http://blogs.rsa.com/rsafarl/phishing-in-season-a-look-at-online-fraud-in-2012/, retrieved on 30th Oct. 2012.

Ryst S. (2006). The Phone is the Latest Phishing Rod, 11th Jul. 2006, http://www.businessweek.com/technology/content/jul2006/tc20060710_811021.htm, retrieved on 2nd Dec., 2011

S1tony (2011). Cookie Monster, https://addons.mozilla.org/en-US/firefox/addon/cookie-monster/, retrieved on 2nd Dec., 2011

Sabherwal R., King W. R. (1992). Decision Processes for Developing Strategic Applications of Information Systems: A Contingency Approach. Decision Sciences, 23(4), pp.917–943.

Sae-Bae N., Ahmed K., Isbister K., Memon N. (2012). Biometric-rich gestures: a novel approach to authentication on multi-touch devices. Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '12). ACM, New York, NY, USA, pp 977-986.

Sahami M., Dumais S., Heckerman D., Horvitz E. (1998). A Bayesian Approach to Filtering Junk E-Mail, Learning for Text Categorization: Papers from the 1998 Workshop. AAAI Technical Report WS-98-05.

Saleem N. (1996). An Empirical Test of the Contingency Approach to User Participation in Information Systems Development, Journal of Management Information Systems, 13(1), pp.145-166

Samoladas I., Stamelos I. (2006). Assessing Free/Open Source Software Quality, http://ifipwg213.org/system/files/samoladasstamelos.pdf, retrieved on 23rd Apr., 2010

Scarfone K., Jansen W., Tracy M. (2012). Guide to General Server Security, http://csrc.nist.gov/publications/nistpubs/800-123/SP800-123.pdf, retrieved on 31st Jan., 2012

Semenov A., Veijalainen J., Kyppö J. (2011a). Analyzing the presence of school-shooting related communities at social media sites. International Journal of Multimedia Security and Assurance (IJMIS), Vol. 1, Nr. 3, pp.232-268

Semenov A., Veijalainen J., Boukhanovsky A. (2011b). A generic Architecture for a Social Network Monitoring and Analysis System. Proceedings of NBIS 2011, Sept. 3-7, Tirana, Albania. IEEE CS, pp. 178-185.

Shahriar H., Zulkernine M. (2010). PhishTester: Automatic Testing of Phishing Atacks, 2010 Fourth international Conference on Secure Software Integration and Reliability Improvement, pp198-207

Sindre G., Opdahl A. L. (2001). Templates for Misuse Case Description. Proceedings of the 7th International Workshop on Requirements Engineering Foundation for Software

Quality, REFSQ'2001, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.9.8190, retrieved on 1st Oct. 2011

Siponen M. (2003). On the Role of Human Morality in Information System Security: From the Problems of Descriptivism to Non-descriptive Foundations. Social Responsibility in the Information Age: Issues and Controversies. Idea Group Publishing. Pp239-254

Siponen M. (2004). A pragmatic evaluation of the theory of information ethics. Ethics and Inf. Technol. 6, 4 (December 2004), pp279-290.

Siponen M., Baskerville R., Heikka J. (2006). A design theory for secure information systems design methods, Journal of the Association for Information Systems 7 (11) pp.725–770.

Siponen M.T., Heikkar J. (2008). Do secure Information System Design Methods Provide Adequate Modeling Support?, Journal of Information & Software Technology (50:9-10), pp. 1034-1053.

SpamAssassin (2001). The Apache SpamAssassin Project, http://spamassassin.apache.org/, retrieved on 12th Aug. 2010.

SpoofGuard (2001), http://crypto.stanford.edu/SpoofGuard/, retrieved on 15th Jan., 2006

Surfthenetsafely (2011a). Advanced Cookie Management in Internet Explorer 6 and 7, http://surfthenetsafely.com/cookie_advanced.htm, retrieved on 10th Jan., 2012

Surfthenetsafely (2011b). Internet Cookie Management in the Firefox Browser, http://surfthenetsafely.com/cookie_firefox1.htm, retrieved on 10th Jan., 2012

Swartz J. (2006). Phishing Attacks Now Using Phone Calls, 26th Nov. 2006, www.usatoday.com/money/industries/technology/2006-11-26-phishing-usat_x.htm, retrieved on 11th Dec., 2011

Syroid T. (2012). Web Server Security, http://www.ibm.com/developerworks/linux/library/s-wssec.html, retrieved on 31st Jan., 2012

Tamrakar S., Ekberg J., Asokan N. (2011). Identity verification schemes for public transport ticketing with NFC phones. Proceedings of the sixth ACM workshop on Scalable trusted computing (STC '11). ACM, New York, NY, USA, pp.37-48.

Tang J., Terziyan V., Veijalainen J. (2003). Distributed PIN Verification Scheme for Improving Security of Mobile Devices. ACM MONET, Special Issue on Security in Mobile Computing Environments, Vol 8, Nr. 2 (April 2003), pp. 159-175.

Tencent Service (2012). Phishing Scams by QQ Video Chat, http://service.qq.com/info/30096.html, retrieved on 2nd Jan., 2012

Trusted computing group (2011a). Mobile Phone Work Group Mobile Trusted Module Specification, http://www.trustedcomputinggroup.org/files/static_page_files/3D843B67-1A4B-B294-D0B5B407C36F4B1D/Revision_7.02-_29April2010-tcg-mobile-trusted-module-1.0.pdf, retrieved on 10th Sep., 2011

Trusted computing group (2011b). Mobile Trusted Module 2.0 Use Cases, http://www.trustedcomputinggroup.org/files/static_page_files/FA751710-1A4B-B294-

D0F1698506A36AE8/TCG%20Mobile%20Trusted%20Module%202%200%20Use%20 Cases%20v1%200.pdf, retrieved on 10th September, 2011

Turner S., Polk T. (2011). Prohibiting Secure Sockets Layer (SSL) Version 2.0, IETF, http://tools.ietf.org/html/rfc6176, RFC6176, retrieved on 11th December, 2012.

Tuunanen T., Rossi M., Saarinen T., Mathiassen L. (2007). A Contigency Model for Requirements Development, Journal of the Association for Information Systems: Vol. 8: Iss. 11, Article 33. Available at: http://aisel.aisnet.org/jais/vol8/iss11/33

Veijalainen J. (2007). Autonomy, Heterogeneity, Trust, Security, and Privacy in Mobile P2P Environments. International Journal of Security and Its Applications, Vol.1. No.1, July 2007, pp. 57-72.

Veijalainen J., Weske M. (2002). Modeling Static Aspects of Mobile Electronic Commerce Environments. Chapter 7 in Advances in Mobile Commerce Technologies, IDEA Group Publishing, 2002, pp. 137-170

Veijalainen J., Hara V. (2011). Towards Next Generation System Architecture for Emergency Services. Proceedings of Information and Security Assurance conference, Aug. 16-18, 2011, Brno, Tzech Republic. Springer Verlag, CCIS Vol. 200, pp. 185-199.

Wang W., Benbasat I. (2012). A Contingency Approach to Investigating the Effects of User-System Interaction Modes of Online Decision Aids, Information Systems Research.

Wang X. Huff P.D. Tjaden B.C. (2008). Improving the Efficiency of Capture-Resistant Biometric Authentication Based on Set Intersection. Proceedings of Computer Security Applications Conference, 8-12 Dec. 2008, pp.140-149.

Warren M., Hutchinson W. (2002). Cyberspace Ethics and Information Warefare, Social Responsibility in the Information Age: Issues and Controversies. Idea Group Publishing. Pp126-134

Webb S., Chitti S., Pu C. (2005).An Experimental Evaluation of Spam Filter Performance and Robustness against Attack, International conference on Collaborative Computing: Networking, Applications and worksharing, 2005. from Georgia Institute of Technology: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.137.5714&rep=rep1&type=pdf, retrieved on 11th Nov., 2010

Wheeler A. D. (2003). Is Open Source Good for Security. Secure Programming for Linux and Unix HOWTO, http://www.dwheeler.com/secure-programs/Secure-Programs-HOWTO/open-source-security.html, retrieved on 20th Oct., 2012.

Wikipedia. (2012). Transport Layer Security, http://en.wikipedia.org/wiki/Transport_Layer_Security, retrieved on 9th Dec. 2012

Wu M., Miller R. C., Little G. (2006a). Web Wallet: Preventing Phishing Attacks by Revealing User Intentions. Proceedings of the second symposium on Usable privacy and security. ACM International Conference Proceeding Series; Vol. 149

Wu M., Miller, R. C., Garfinkel, S. L. (2006b). Do Security Toolbars Actually Prevent Phishing Attacks? Conference on Human Factors in Computing Systems

Yardley G. (2007). Delete Flash Local Shared Objects, http://objection.mozdev.org/, retrieved on 11th Dec., 2011

Yang Y., Pedersen J. O. (1997). A Comparative Study on Feature Selection in Text Categorization, ICML '97 Proceedings of the Fourteenth International Conference on Machine Learning, 412-420, San Francisco, USA

Youn S., McLeod D. (2007). Efficient Spam Email Filtering using Adaptive Ontology. Proceedings of International Conference on Information Technology, 249-254, Las Vegas, USA

Zhang L., Zhu J., Yao T. (2004). An Evaluation of Statistical Spam Filtering Techniques, ACM Transactions on Asian Language Information Processing (TALIP), 3(4), 243-269.

Zhang Y., Egelman S., Cranor L., Hong J. (2007). Phinding Phish: Evaluating Anti-phishing Tools. Proceedings of the 14th Annual Network & Distributed System Security Symposium, San Diego, CA, USA

ZoneAlarm (2012). The Dark Side Of Social Media: How Phishing Hooks Users. ZoneAlarm Blog, http://www.zonealarm.com/blog/index.php/2011/07/how-phishing-hooks-users, retrieved on 6th Dec. 2012.

Zorz Z. (2011). Sony Online Store Hacked and User Information Published, http://www.net-security.org/secworld.php?id=11074, retrieved on 23rd Oct., 2011

Study 1

Li L., Helenius M., Berki E. (2007). Phishing-Resistant Information Systems: Security Handling with Misuse Cases Design, *Proceedings of Berki, E., Nummenmaa, J., Sunley, I., Ross, M. & Staples, G. (Eds) Software Quality in the Knowledge Society*, SQM 2007. Tampere, Finland,1-2 August 2007, pp. 389-404.

Phishing-Resistant Information Systems: Security Handling with Misuse Cases Design

[1]Linfeng Li, [2]Marko Helenius, [2]Eleni Berki

[1]F-Secure Corporation, Tammasaarenkatu 7, 00180 Helsinki, Finland
linfeng.li@f-secure.com

Department of Computer Sciences, University of Tampere, Finland
{cshema, eleni.berki}@cs.uta.fi

Abstract

The spread of phishing intensifies the need for well-defined security requirements in the design of an information system. Phishing goes on increasing, especially in the e-services domain, even though a variety of prevention methods have been developed and used against it. Phishing attacks compromise the software quality features of a system. In our study, we focus on how to prevent phishing attacks with the misuse case method from a system design perspective. After presenting the phishing attack techniques and the related threats, we introduce and evaluate three kinds of the existing phishing prevention methods. As an evaluation result, we express our support and give a brief introduction to the misuse case method; we subsequently construct an example scenario to illustrate the method's application in the phishing prevention domain. After the discussion on phishing prevention based on the misuse cases identification , we conclude that it is possible to cater for phishing attacks at the system design level, by considering design quality features that ensure system's security.

1.0 Introduction

Software quality [1, 2, 3] is not a set of essentially wanted and desirable features that can be added to a system after its realisation; software quality features [1, 2, 4, 5, 3], and especially those that deal with functionality, are planned and designed from the very initial phases of the software development lifecycle [1, 2, 4, 5]. Dealing with system functionality, ISO 9126 defines functionality as a set of attributes that bear on the existence of functions and their specified properties [1, 2, 4, 5, 3]. The functions are those that satisfy stated or implied needs; therefore, they must be and prove to be *suitable, accurate, secure* and with certain *interoperability* features [see e.g. 3]. Evidently, many existing information systems do not bear these characteristics no matter what and how software development methods, tools and quality models are used [see e.g. 1 and 2].

Design quality issues regarding system's functionality (security in particular) and usability (operation in particular) are still not holistically considered by information systems designers [6]. As a result, an inaccurate and not precisely designed system will not bear the security required. In web-based information systems, for instance, virtual identities are required for interaction [7]. Virtual *identity theft* is a frequent phenomenon, which is not new; it has, though, become a problem haunting people's daily lives. Identity theft is a very general term, which can further be categorised into many sub-classes based on the media it takes advantage of when stealing the identity used for communication. Identity thieves can profoundly exploit a number of insecure transmission tools including telephone, mail, email and various websites. Apparently, the most convenient and significant technique to steal someone's identity is using the Internet, that is email and website; this technique is called phishing [8].

In phishing attacks the aim is to steal the users' confidential information (e.g. credit card number, password, PIN code etc.) by *social engineering* and *technical subterfuge* [9]. According to the definition from Anti-Phishing Work Group (APWG), these concepts of phishing are described as follows: *"Phishing attacks use both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials. Social-engineering schemes use 'spoofed' e-mails to lead consumers to counterfeit websites designed to trick recipients into divulging financial data such as credit card numbers, account usernames, passwords and social security numbers. Hijacking brand names of banks, e-retailers and credit card companies, phishers often convince recipients to respond. Technical subterfuge schemes plant crimeware onto PCs to steal credentials directly, often using Trojan keylogger spyware."*

Those emails and websites, used - or rather abused - by social engineering and technical subterfuge, impersonate the authentic ones, normally including the same website layouts and logos, and even similar domain names. All these basic design characteristics and virtual features are imitated so well that the majority of the end-users can hardly distinguish between them. In addition, due to the lack of effective fraud detection techniques, a great number of inexperienced email and website end-users' identities are compromised under the threat of theft and fraud.

According to information obtained from the records of APWG, the number of phishing attack reports reached 18480 in March 2006 [8]. Although there is no economy loss specified in the report, one could (not!) imagine how much that could be. Based on a report by Javelin Strategy and Research on April 2006, the economy loss reached 20 billion [10]. Most likely, the veracity of the figure might be argued. Nevertheless, phishing seriously challenges and collapses the trust to electronic commerce and e-services security. Less and less users feel secure and, as a result of this insecurity in e-services, they might stop using otherwise convenient online services, since they are not sure whether their credentials are in danger. Therefore, the questions on (i) how to identify the fraud, and (ii) how to design and build a reliable and secure environment for business transactions have become the most imperative requests of this research field.

So far, there has been some significant, albeit not adequate, progress in this field that has taken into account both the clients' considerations and the servers' design quality features. On the client side, there have been more than eighty (80) types of user-centred applications developed; these are automated techniques such as browser toolbars and plug-ins. Meanwhile, more and more researchers on the topic of security realise the need for improving server security, in order to holistically protect against phishing by considering both the client and the server. Regarding the latter, there are two typical outcomes: One is the *phishing email filter* on the email server developed by Carnegie Mellon University [11]. This filter was designed with *learnability* as the main software quality feature in mind; this means that the filter contains *self-learning capacity* and intelligence to detect and identify the phishing emails. The other one is a *light weight trust architecture* designed by Massachusetts Institution of Technology [12]. The key software quality feature of this new platform is the light weighted self-certificate for *verification*; this design quality characteristic can verify each other's identities in private communication situations.

Undoubtedly, all these research efforts and outcomes are quite helpful for fighting against phishing attacks. However, the performance (or efficiency) of these tools is not satisfactory [13] and, as far as we know, no research focuses on how to build a *phishing-proof* system from the system's design point of view. In this paper, we look at conceptual design and technical design issues for such a phishing-proof system and explain how, after the requirements analysis stage, an information system can be designed, validated and documented against phishing, with the *misuse case method.*

The rest of the paper is organized as follows: In the second chapter the basic classification of phishing attacks are presented. Subsequently, a summarised description of existing approaches against phishing attacks is provided and a brief analysis of them is given, paying particular attention to clarify their functionality, reliability, usability and efficiency (performance) [see e.g. 3]. After reviewing and commenting on their strengths and weaknesses, we introduce the misuse-oriented methods and give a group of sample misuse cases for preventing from phishing attacks. In so doing, we aim at the demonstration of the method's applicability and design strengths. For future research we underline its potential to serve as a proof-phishing thinking and technical tool. Its utilisation could advance conceptual understanding and cognitive thinking for metamodelling and could enrich software quality assurance techniques in the early analysis and design phases.

All in all, we believe that our work in this paper can serve as a reference guide to software developers and any e-people/end-users who want to find a set of simple, understandable and practical new knowledge on phishing and anti-phishing techniques and concepts. This concise, recent collection of updated, subsequently simplified and classified information could also serve as a safety guide for novices and inexperienced users while interacting in virtual communities.

2.0 Classification of Phishing Attack Methods

In order to assume proper, or at least reasonable, misuse cases when defining the security requirements in the design phase it is necessary to have access to updated information and possess suitable design knowledge. Otherwise, there is a need to collect, analyse, understand and finally classify the existing phishing attacks. There are various classifications available. Helenius [14], for instance, analysed and classified the phishing attacks based on the different techniques used for phishing. These include the following: (i) user request, (ii) redirection, (iii) user-end hijacking, (iv) banking side hijacking, while the "co-operation" of some of these techniques is also possible.

We, hereby, define our own classification criteria considering different entities that apparently are the current popular target of phishing attacks. These are: *a. online banking users, b. transport media, and c. financial institutions' servers.* Before presenting our full classification on the latter in section 3 of this paper, we proceed to a brief description of these entities and domains below. The following sections 2.1 - 2.3 provide, among other information, an analysis on the threats and vulnerability of the entities in these domains and on how security is compromised.

2.1 Online banking user side attacks

Online banking user side phishing attacks are very common, compared with the other two target entities. The reason why phishers prefer to aim at end-users is that they can be easily cheated or spoofed without using professional techniques. These attacks are very successful regardless the level of experience or inexperience that the end-users possess.

The most common and easiest phishing on the client side is the *man-in-the-middle* attack. Phishers mislead and spoof users with emails or phishing pages. When victims reply to them with confidential information enclosed, the information is collected by the phishers. After that, phishers are able to log into the victims' accounts. According to the sources of emails, phishing emails could be classified into two different types. (i) The first one is from the phony banking institutions. Usually, these emails come with an announcement like the next: there is bait, which says there is something wrong with the account, or a bonus is distributed, or a competition takes place and so on. Credentials are asked for confirmation. This may be elusive even for experienced users, as one would not like to miss something financially important.

(ii) The second type of the spoofing emails seems to be coming from the friends or relatives of victims. This is called *social phishing* [15]. This kind of phishing email takes advantage of the trust relationships between people and their acquaintances. Because of the *trust relationship*, victims may be convinced. Moreover, this kind of phishing emails spread much faster than in the first type. However, deceptive emails are not the end. To harvest confidential information, there are two basic ways. One is via email itself, and the other is by offering an URL of a fraudulent website, which looks the same as its authentic website (with the similar logo, website roadmap, page layouts, and domain name). Apparently, all these are difficult to test and identify; testing for security and authenticity is another research milestone to be reached in software development.

A more technical attack resorts to ActiveX or other explorer plug-in techniques. The most notorious plug-in is a *keylogger*, which can record the key pressing and send the recorded key strokes to the attacker stealthily. Obviously, these types of attacks are more dangerous, since the potential victims are not aware of the attack. Moreover, due to the fact that end-users are not necessarily equipped with that specialised computer science or IT knowledge, it is extremely difficult for novice and occasional end-users to detect keyloggers. Similarly, another technique named *screen logger*, also collects the movements of a mouse cursor. For the customers using on-screen virtual keyboard, screen logger exposes them to phishing.

Undoubtedly, these malicious plug-ins are not the only form of technical phishing attacks on the client side. The traditional malware can also be employed here. When a computer virus, worm or software Trojan horse controls an operating system, any confidential information can be divulged and the whole appearance of the system can be changed.

2.2 Network transport media attacks

Hacking *network transport media*, literally, means that attacks are launched towards network transport media, such as routers, switchers, and especially DNS (Domain Name Server) servers. These core network infrastructures are valuable for Internet criminals. It can be a nightmare, if phishers control some of these transport media. In practice, phishers aim to redirect user's TCP/IP (Transmission Control Protocol /Internet Protocol) requests to a preserved forged website by hijacked DNS servers [16]. These compromised DNS servers are still difficult to detect manually.

2.3 Phishing attacks on the financial service side

Fortunately, there is a limited number of reports about the fact that the servers of financial institutions have been compromised so far. However, hacking online banking servers is still a possible threat. Most of the popular server side attacks can also be used by phishers, including SQL injection attack and PHP injection attack. These attacks are still a risk for wrongly configured servers.

3.0 Prevention against Phishing Attacks

On the other hand, the prevention against phishing can also be classified with the same rules as used for classifying phishing attacks. The following sections provide our classification with a commentary on the basic quality features that can be found in each of them.

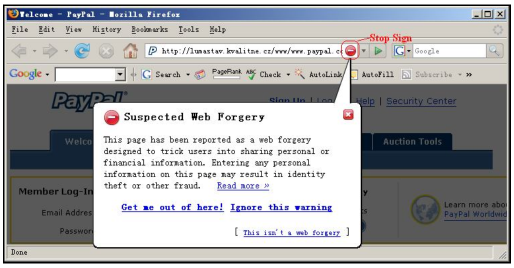
3.1 Client side applications

Nowadays, most prevention methods concentrate on the most vulnerable client side. For example a prevention application is installed on a personal computer to assist the user while identifying fraud attempts. In general, the most widely used application to fight against phishing is a *browser toolbar*, which is embedded into an Internet browser. According to the toolbar implementation architecture, toolbars can be divided into two types: One is based on the client-server structure, while the other works only on a personal computer without any anti-phishing server.

3.1.1 Client-server structured applications

These kinds of toolbars are normally deployed both on a client machine and on a server. The toolbar requests the server for regular updates and maintenance frequently. In this regard, this type of toolbars is normally developed and released by commercial companies, such as Google, Netcraft and Microsoft.

For example, Google Safe Browsing (Figure 1), is a part of the Google toolbar extension for Firefox. It is able to alert users, when the web page visited is judged as a fraudulent one. Google Safe Browsing blocks the visit of web pages by using a blacklist. According to the introduction on the toolbar's download page, the blacklist is generated and maintained by a server hosted by Google [17]. In order to determine the authenticity of a web page, the toolbar sends the visited URL to the server, when a user tries to browse any web page. If the URL is considered as a deceptive one, the toolbar will stop the user's activity and give appropriate advice (e.g. stop visiting the fraudulent web site).

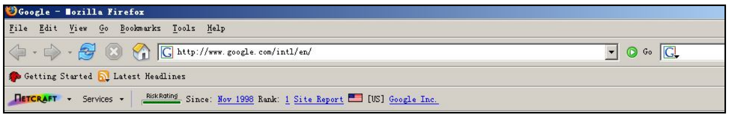


Suspected Web Forgery

This page has been reported as a web forgery designed to trick users into sharing personal or financial information. Entering any personal information on this page may result in identity theft or other fraud. Read more »

Figure 1. Google Safe Browsing toolbar, when a phishing page is detected.

The mechanism of the Netcraft Anti-Phishing toolbar (Figure 2) is similar to the Google's one. The Netcraft toolbar also communicates with the Netcraft site's database [18] and obtains a blacklist. Moreover, the toolbar offers extra information concerning the page visited. For example, when a user visits a website, he or she can easily find the risk rating, the established time, the Internet rank and the location of the visited website.



Google - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.google.com/intl/en/
Go
Getting Started
Latest Headlines
NETCRAFT
Services
Risk Rating
Since: Nov 1998 Rank: 1 Site Report
[US] Google Inc.

Figure 2. Netcraft anti-phishing toolbar

3.1.2 Independent applications

Compared to client-server toolbars, independent toolbars identify a deceptive website based only on the data stored on a personal computer. The general mechanism of this type of toolbar is like this: After a web page is downloaded to a personal computer the toolbar will compare the characteristics of this web page to with the previously saved data, e.g. domain name, image harsh and so on. If any differences are found, the toolbar will warn the user and give some suggestions.

SpoofGuard, shown in Figure 3, is the outcome of a study that took place at Stanford University. The toolbar is compatible only with Microsoft Internet Explorer. Compared with previous two toolbars, this one uses a kind of *whitelist*: the browsing history of the Internet Explorer. There are three buttons on the toolbar, which are next briefly described: one acting as 'traffic light' for indicating the security status of the visited page, another as 'hammer' for configuring the settings, and a third acts as 'crossing' for removing all data collected by SpoofGuard, that is image hashes and password hashes. SpoofGuard is able to identify fraudulent web pages by checking the browsing history as well as other information, including domain name, URL, email, password field, image and links on the visiting page [19]. Moreover, SpoofGuard can alert a user, when he/she submits the same ID and password to different web pages. When a warning shows up, two choices are given: to 'continue' or 'stop visiting'.

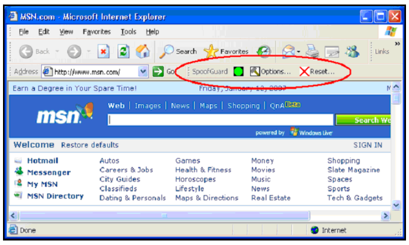


Figure 3. SpoofGuard is circled in red

Anti-phishing IEPlug is a whitelist based on Microsoft Internet Explorer plug-in against phishing web pages, implemented by the authors of this paper at the Department of Computer Sciences at the University of Tampere [20]. This plugin is able to show the *certificate and authority* of web pages containing a password field. The certificate is shown after a user has added the domain name to the whitelist [20]. The toolbar offers an interface for maintaining the domain names including adding, editing and removing with accuracy. Only a limited number of users can add domain names to the list, while administrators can maintain the list. The toolbar also has a limited capability to warn about fraudulent web pages. The detection is based on a keyword search from the address(es) visited.

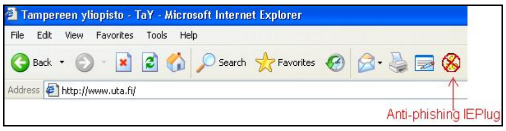

Tampereen yliopisto - TaY - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.uta.fi/
Anti-phishing IEPlug

Figure 4. IEPlug button on the toolbar of IE browser

3.2 Server side applications

The term 'server side' here does not only mean the servers in financial institutions. Instead, it also includes servers running on the whole network.

One of the available applications focused on the server side is, as already mentioned in the Introduction part, the phishing email filter, which is developed in Carnegie Mellon University [11]. This application is deployed on the email servers. The basic detecting mechanism is similar to the traditional spam filters. However, the most significant software quality feature of the filter is that it is equipped with the capacity of learning (learnability). This means that the application can improve its intelligence-based detection capacity during and after many emails detection.

This is undoubtedly a beneficial characteristic utilised for a phishing attack. Because phishing emails do not follow any specifications or rules, to effectively detect them is a big challenge for the non-intelligent applications.

3.3 Analysis of the existing anti-phishing applications

These anti-phishing applications are helpful in fighting against phishing. However, the performance of these applications is not so satisfactory. According to the results of an evaluation test from Carnegie Mellon University [21], some toolbars are not that ideal to realize the goal of phishing prevention. In the test, two (2) out of five (5) toolbars with best evaluation score can only successfully identify eighty per cent (80%) of the pages that phishing takes place. The other toolbars have even worse performance, since they are just able to detect only forty per cent (40%) of these pages.

In fact, as discussed in the previous sections, phishing is not the same as traditional viruses or other attacks. Attempting a cognitive association, we may state that phishing actually takes advantage of the security breaches in end-users' thinking model. Examining this further, we construct the following simple scenario:

1. A user wants to confirm the password of his or her bank account.
2. The user logs into the system, and then gives what the phishing web page asks for.
3. After that the user clicks the submit button on the page.

These three simple steps can easily be conducted by any people. Usually, people are familiar with a similar scenario, that is when they want to change their bank account password in the bank's webpage. Moreover, the user interface of the bank's webpage and the phishing webpage may be the same.

In the real world, the bank users need to contact bank clerks, who can be identified from their appearances and the office in the bank place. In this case, the bank users just need to verify who is managing their accounts. However, in the virtual world, this trust relationship between the service and the customer is not the same any more. The e-Customers make their e-transactions with the help of web pages, the Internet and computers. The appearances of these objects can easily be forged. Therefore, if the bank or any other e-users keep following the same thinking model to check the e-service's outlook (appearance), the breach is obvious: the users are exposed at the unawareness of the identification in online banking and other online services.

Even though the existing toolbars can assist users' safe browsing, they hardly change the mental model in users' minds. Thus, the designers of phishing prevention applications should find solutions to enhance functionality and reliability of the IT services from the system design point of view. At design level, the software and system design explicitly stated requirements normally include the description the demands and conditions for the final system's secure transaction procedure. Moreover, the design style and design method could cater for time behaviour and resource behaviour and enforce users to behave in a more secure and stability-oriented mental model, for example by using the one-time password. Of course the question of how to design a robust system to prevent from phishing requires catering for more than for the one-time password. Information systems should be planned and analysed well at the requirements planning and analysis stage. At this stage, designers could elicit essential security requirements against phishing. In the next section, we introduce the *misuse case method* for phishing-resistant information systems.

4.0 Misuse-oriented Phishing Prevention

In this chapter, we introduce the misuse case method to prevent phishing from the system design perspective. First, the misuse case methodology is explained. After that, there is a system design example given to illustrate how to find the requirements of fighting against phishing. Finally, a brief analysis of the misuse case method is given.

4.1 Misuse case methodology

The first time that a misuse case was used for security requirements elicitation was achieved by the co-operation between the Norwegian University of Science and Technology and the University of Bergen [16]. According to the article, the misuse case was based on the analysis of use cases at the requirements gathering stage. In brief, a designer is asked to impersonate a misuser to abuse every use case (misuse case). When a threat is defined by a misuse case, the corresponding countermeasures can be identified and employed to prevent from the threat in the future. Of course, it is possible that there remain potential vulnerabilities in the countermeasure. Therefore, a next round misuse case design against the countermeasure(s) is required, until there is no possible vulnerability found.

In summary all the steps of the methodology of the misuse cases [16] can be induced in the following six (6) steps:

a. *Design the use cases of the system;*
b. *personate a misuser, who intends to compromise the system;*
c. *design the misuse for a specific use case;*
d. *find a countermeasure for a misuse case;*
e. *judge whether the countermeasure is vulnerable; if yes, go to step c, otherwise go to the next step;*
f. *find whether there is possible vulnerability or misuse; if yes, go to step c, otherwise security requirements elicitation ends.*

In the following section we provide an example scenario to demonstrate and clarify the use of the above.

4.2 Misuse cases to design an online music purchase website

In order to demonstrate the value of the misuse case method, let us consider an e-shop MusicBox for music purchasing with an available website. For the purpose of extracting reasonable misuse cases for the design rationale, the description of the normally available e-services on the website is given below.

The MusicBox is an online music products provider. Its e-services are designed with functionality and availability in mind to include, among other online services, free listening of a few favourites, online latest music clips listening, and vending music products and so on. This website comprises several functional and usable components: (i) a web browser, (ii) client application, (iii) front-end server, (iv) content database and (v) credit card server (see Figure 5).

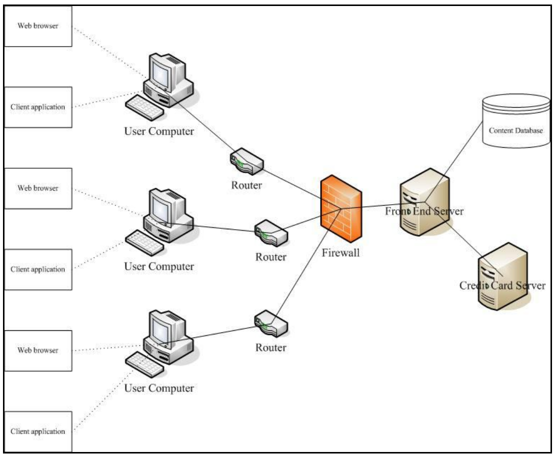


Figure 5. Architecture overview of a music purchase website

A web browser, such as Internet Explorer or Firefox, runs on the webpage users' personal computers and users can also access the website of the MusicBox via the browser. The functions offered on the website contain listening to the music clip promotions, opening a customer account and logging into an account when purchasing music products. Only after a user creates an account in the e-shop of the MusicBox, he/she is able to purchase the music products. In order to buy music products, valid credit card details are required. When valid credit card information is given, credit card server will handle the credit card transaction. Afterwards, the content database makes the music products available, which can also be downloaded to the user's personal computer. The client application, a decode player, can decode and play the music file.

Herein, the use cases are utilised to design the misuse case(s). The use cases, thus, are not listed within any industrial template but within our framework of thinking. In the following lines UC is an abbreviation of Use Case, and the number is a simple sequential indicator for the use cases listing. In summary, the use cases for the MusicBox described in the previous paragraph, will be:

UC-1. Any user can listen to the music clip promotions.

UC-2. The user who wants to create an account in MusicBox should offer a valid email address, password and a user name.

UC-3. The email for confirming the account is sent to the user's registered email address.

UC-4. After confirmation, the user's account is activated. Then, the user is able to purchase the music products.

UC-5. A user can purchase a music product, after logging in.

UC-6. A user should provide valid credit card number for purchasing.

UC-7. Credit card server should delete users' credit card information after successful purchase.

UC-8. A confirmation email, 1) to notify user about successful purchase, should be sent to user's email address left, when he or she has created the account. A valid link to download the music product is added into the account profile; 2) to notify the user who does not use the account for a certain long time.

UC-9. User can download the music from the link in his or her account profile.

UC-10. After downloading completes, a user can display the music via client application which is downloadable from the MusicBox free of charge.

UC-11. The downloaded music from the MusicBox cannot be displayed by other music players.

UC-12. The purchased music product can be downloaded as many times as needed by the account owner.

After listing the Use Cases one might ask who could be a *future system misuser*. Possible system misusers in the future could be code crackers and MusicBox users who want to re-sell or share the music with others. After the use cases are elicited, misuse cases can then be designed. Herein, the Sindre's template [16] is applied. One sample of misuse case is given in Table 1.

Table 1: Misuse cases sample A

Name:	Sending the email to confirm the use of one account
Case ID:	MC-1
Summary:	A cracker can obtain the account name and password by sniffing plain messages.
Author:	******
Date:	2007-3-10
Basic path:	1. A cracker can obtain the account name and password from sniffing the communication of the server.
Alternative paths:	1. A cracker can harvest account name and password from a forged website. 2. Man-in-the-middle attack can be used for obtaining this information 3. Compromising the server
Capture points:	1. Password is changed 2. Forged site has been detected by the toolbar
Extension points:	1. Keep sniffing communication
Trigger:	This can happen at any time.
Assumptions:	1. The server and forged website are ready. 2. The *MusicBox* server security capacity is not sufficient. 3. A number of customers created their own accounts
Preconditions:	1. The communication is not encrypted.
Worst case threat:	1. Launch another social phishing attack 2. The music can be downloaded freely.
Prevention guarantee:	1. SSL or TSL is used to protect secure communication 2. Login history offered to check the account history

	3. Frequently change password
Detection guarantee:	1. Check login history
Related business rule s:	1. The purchased music product is listed in the account profile 2. The purchased music product can be downloaded for several times. 3. Users can access the account with the account name and password.
Potential misuser Profile:	1. Sufficient computer skills
Stakeholders and Threats:	*MusicBox*: users complain that the system is not secure to be used. Customers: they lose their money.
Scope:	Entire business
Abstraction level:	Misuser goal

Still, the system is not completely phishing resistant. For example, when a user has not used his or her account, the system should notify the user to confirm its account. In this case, it is also possible to be spoofed. Its misuse case can be given as follows.

Table 2: Misuse case sample B

Name:	Steal password and account name by sniffing plain messages
Case ID:	MC-2
Summary:	A phisher impersonates the *MusicBox* staff and sends a email to make a user confirm the use of his or her account.
Author:	******
Date:	2006-11-20
Basic path:	1. A phisher can send the email by any email client, such as Outlook.
Alternative paths:	1. A phisher may impersonate a friend of the target, and pretend to remind the victim by sending an email with “official” link to confirm the use of the account 2. A phisher may ask for the account confirmation so as to update the security status of it.
Capture points:	1. Using a specific customized certificate to identify the authenticity of the email source. 2. Pausing the processing of suspicious business transactions. 3. Using confirmation code for each transaction.
Extension points:	1. Guess the customized certificate
Trigger:	This can happen at any time.
Assumptions:	1. The server and forged website are ready, and successfully avoid to be detected. 2. The targets’ emails are collected. 3. The forged emails are sent to valid the targets’ email boxes.
Preconditions:	1. There should be a regulation to remind users to keep their account activated.
Worst case threat:	1. Users do not believe even the authentic emails from *MusicBox*.

	2. More users quit to use the *MusicBox*.
Prevention guarantee:	1. The customized certificate to identify the authenticity of the email source has been decided when a user register. 2. The requested confirmation code for each transaction varies all the time.
Detection guarantee:	1. Checking the transaction history
Related business rules:	1. The purchased music product is to be confirmed by the user itself. 2. Every user account should be activated.
Potential misuser Profile:	1. Awareness of building the website, sending fake emails with the URL of the forged sites.
Stakeholders and Threats:	*MusicBox*: users complain that the system is not secure to be used, and there are less and less users using *MusicBox*. Customers: some of their accounts become the tool of social phishing.
Scope:	Entire business
Abstraction level:	Misuser goal

The misuse case sample B is also not phishing resistant. Phishers may launch the second round of man-in-the-middle attack to ask the confirmation code, after they get the victims' account name and password. In this case, requirement engineers and system designers need to create another misuse case, and then recursively repeat the previous steps until there is no more possible breach to be found.

4.3 Analysis on misuse case method against phishing

From the example presented in Table 1, we can easily find that the misuse case method is a system-oriented, and rather design-centred, method. This means that different systems can have different misuse cases. Due to this feature, the misuse cases have to be deductively designed for every information system. The systems designed with misuse cases could, thus, be *phishing-resistant*. Since misuse cases are part of the design documentation we propose that existing systems could be re-designed and re-engineered for improving their security without great financial cost. Misuse cases identification in legacy systems, in the way we demonstrated it earlier, could be a cost effective quality solution to provide security. Moreover, the misuse case method could be helpful for validating security requirements when designing new information systems without considering the budget expenses for the future system updates.

On the other hand, the quality and the quantity of misuse cases prompt to adequate knowledge and experience on system design and testing phases by the designers and testers. In order to find out the possible phishing attempts, or at least as many as possible, the designers and testers should be familiar with most of or all system vulnerability issues that exist. Particular attention should be paid in the essence and philosophy of each type of attack. If system designers and testers know little about phishing, software quality assurance techniques alone can not offer much. Regarding system security, for instance, a worthy misuse case to simulate a phisher cannot be provided by an inexperienced designer, even though the misuse case's recursive procedure may help.

5.0 Conclusions

Software-based systems for e-transactions are currently facing many unresolved software quality aspects regarding their operability, security and efficiency. Security requirements in particular are not well-stated in the requirements gathering and analysis stages. This might be the result of the inexperienced designers decisions, and not adequate knowledge on test and use cases. On the other hand, impacts on software quality might also be factors such as lack of reliable, mature and understandably suitable (for the particular system's domain) misuse cases and test cases. Consequently, the system's fault-tolerance, recoverability and functionality are compromised when the information system becomes a target for phishing.

In contributing to an improved understanding of the phishing and anti-phishing methods that will be useful to researchers and especially practitioners when they deal with system design quality features, we proceeded as follows: We provided a critical review and comparative analysis of the features of the existing phishing techniques and their prevention methods. In so doing, we provided a broad classification with the distinct features of the phishing and anti-phishing activities under examination. Phishing itself, a semantically serious attack on organisational, social, personal and interpersonal information systems, is very difficult to prevent. The purpose of each phishing technique, (e.g. in email context) may vary randomly, while the existing prevention methods are fixed. This natural defect has already restricted the overall performance and detection effectiveness of the prevention methods significantly, regardless the continuous evolution of phishing techniques.

Considering this special case, we may conclude that the more is known about limitations of existing applications against phishing, the more applicable a method on misuse case can be. Therefore the design quality features can offer reliability to adequately prevent system users from phishing. The drawbacks, however, of the misuse cases in design are also obvious. In particular, this method requires designers with ample experience in system design and system security. Our future research efforts will concentrate on (i) defining further software quality criteria for a design architecture that contributes to anti-phishing prevention; and (ii) making misuse cases more valuable (and perhaps more reusable) and more formal in order to be used as a quality assurance technique in the validation and verification of a system examining its design.

At system's design phase, testing of software design architectures could provide early feedback on the robustness and vulnerability of the domain-specific system architecture including information on both client and server identities, pitfalls and possible threats. In software development early feedback on software design quality features is a software process improvement issue, in practice. Our ongoing research deals with this and other software process improvement issues searching to provide an answer in the future research study. Other research efforts concentrate on advancing theoretical knowledge on the way system design and software architectures provide safety to e-people interacting in virtual communities worldwide.

6.0 References

1 Berki, E. Examining the Quality of Evaluation Frameworks and Metamodeling Paradigms of Information Systems Development Methodologies. Book Chapter. Duggan, E. & Reichgelt, H. (Eds) *Measuring Information Systems Delivery Quality.* Pp. 265-289, Idea Group Publishing: Hershey, PA, USA, Mar 2006.

2 Berki, E., Georgiadou, E. and Holcombe, M. (2004). Requirements Engineering and Process Modelling in Software Quality Management – Towards a Generic Process Metamodel. *The Software Quality Journal*, 12, pp. 265-283, Apr. 2004. Kluwer Academic Publishers

3 International Organisation for Standardisation (1991). Information Technology-Software product evaluation: Quality Characteristics and Guidelines for their use. ISO/IEC IS 9126. Geneva: ISO.

4 Georgiadou, E., Siakas, K. and Berki, E. (2003). Quality Improvement through the Identification of Controllable and Uncontrollable Factors in Software Development. Messnarz, R. & Jaritz, K. (Eds) EuroSPI 2003: European Software Process Improvement, EuroSPI 2003 Proceedings, 10-12 Dec 2003, Graz, Austria. Pp. IX 31-45. Verlag der Technischen Universität: Graz.

5 Berki, E. and Georgiadou, E. (1996). Towards resolving Data Flow Diagramming Deficiencies by using Finite State Machines. I M Marshall, W B Samson, D G Edgar-Nevill (Eds) *Proceedings of the 5th International Software Quality Conference*. Universities of Abertay Dundee & Humberside, Dundee, Scotland, Jul 1996.

6 Berki, E., Isomäki, H. and Jäkälä, M. (2003). Holistic Communication Modelling: Enhancing Human-Centred Design through Empowerment. Harris, D., Duffy, V., Smith, M., Stephanidis, C. (Eds) *Cognitive, Social and Ergonomic Aspects, Vol 3 of HCI International*, 22-27 Jun 2003, University of Crete at Heraklion, pp. 1208-1212, Lawrence Erlbaum Associates Inc.

7 Jäkälä, M. and Berki, E. (2004). Exploring the Principles of Individual and Group Identity in Virtual Communities. Commers, P., Isaias, P. & Baptista Nunes, M. (Eds) Proceedings of the *1st IADIS Conference on Web-based Communities*. Mar 24-26 Lisbon. Pp 19-26. International Association for the Development of Information Society (IADIS): Lisbon.

8 Anti-phishing organization. www.antiphishing.org (visited January 2007)

9 Wikipedia. www.wikipedia.com (visited November 2006)

10 Javelin Strategy and Research. http://www.javelinstrategy.com/ (visited November 2006)

11 Kumaraguru P., Rhee Y., Acquisti A., Cranor L., Hong J., and Nunge E.. Protecting People from Phishing: The Design and Evaluation of an Embedded Training Email System. CyLab Technical Report. CMU-CyLab-06-017, 2006.

12 Chau, D.. Prototyping a Lightweight Trust Architecture To Fight Phishing. MIT undergraduate projects, May 2005. http://theory.lcs.mit.edu/~cis/theses/chau-uap.pdf (visited December 2006)

13 Zhang Y., Egelman S., Cranor L., and Hong J.. Phinding Phish: Evaluating Anti-Phishing Tools. Carnegie Mellon University, CyLab Technical Report. CMU-CyLab-06-018, 2006, http://www.cylab.cmu.edu/default.aspx?id=2255, (visited November 2006).

14 Helenius M.. Fighting against Phishing for OnLine Banking Recommendations and Solutions. Papers and Presentations of the15th Annual EICAR Conference "Security in the Mobile and Networked World" . pp. 252-267, May 2006.

15 Jagatic T., Johnson N., Jakobsson M., and Menczer F.. Social Phishing. Indiana University Stop-Phishing group. December 2005. http://www.indiana.edu/~phishing/social-network-experiment/phishing-preprint.pdf. (visited September 2006).

16 Sindre G. and Opdahl A.. Templates for Misuse Case Description, 2001, http://www.ifi.uib.no/conf/refsq2001/papers/p25.pdf. (visited November 2006)

17 Google Toolbar. http://www.google.com/support/firefox/bin/static.py?page=features.html&v=2.0f (visited January 2007)

18 Netcraft anti-phishing toolbar. http://toolbar.netcraft.com/ (visited November 2006)

19 SpoofGuard. http://crypto.stanford.edu/SpoofGuard/ (visited November 2006)

20 IEPlug-in. http://www.cs.uta.fi/~ll79452/ap.html (visited December 2006)

21 Cranor L., Egelman S., Hong J., and Zhang Y.. Phinding Phish: An Evaluation of Anti-Phishing Toolbars. CMU Technique Report CMU-CyLab-06-018.

Study 2

Li L., Berki E., Helenius M. (2011). Evaluating the Design and the Reliability of Spam/Phishing Content Filtering Performance Experiments, *Proceedings of Dawson, R., Ross, M., Staples, G. (Eds), Global Quality Issues, SQM 2011*, Leicestershire UK, 18 April 2011, pp.339–357.

Evaluating the Design and the Reliability of Spam/Phishing Content Filtering Performance Experiments

[1]Linfeng Li, [2]Eleni Berki, [3]Marko Helenius

[1]F-Secure Corporation
Tammasaarenkatu 7, 00180 Helsinki, Finland
linfeng.li@f-secure.com

[1, 2]School of Information Sciences, University of Tampere,
Kanslerinrinne 1, Pinni B, FI-33014, Tampere, Finland
{linfeng.li, eleni.berki}@uta.fi

[3]Department of Communications Engineering, Tampere University of Technology,
Korkeakoulunkatu 1, FI-33720 Tampere, Finland
marko.t.helenius@tut.fi

Abstract

Unsolicited spam messages, delivered with fraudulent information, disturb people's web browsing and bring huge risks because personal confidential information and secure web browsing are compromised. Even though a variety of spam/phishing filters are deployed, the performance of these varies, especially regarding the quality of content filtering. When designing the performance experiment and criteria for content filtering, researchers usually focus on different angles. The authors compare and contrast the performance of content filtering from the perspective of information retrieval. Further, the authors critically review the existing research results in a meta-evaluation study and discuss how to design a *reliable* performance evaluation experiment on content filtering features. We conclude that enhanced reliability results in increased security for email communication and web browsing.

1.0 Introduction

Unsolicited and fraudulent emails, continuously spreading widely and rapidly, bring risks for abusing online identities. Additionally, there is a waste of considerable time and computing resources needed to pre-process and prevent from these malicious messages. Various popular anti-spam/phishing technologies [1] have been implemented and deployed on both server and client sides, including *blacklist based detection, whitelist based detection, content filtering* and the list can go on. Blacklist based detection, for instance, checks the properties of the incoming emails, e.g. IP address, against the preserved data in a frequently maintained and updated server searching for malicious messages. The whitelist based detection, on the other hand, works differently by storing data such as IP address or domain names, allowed to be delivered.

Background: Besides these list-based spam/phishing detections, the anti-spam/phishing technologies based on content filtering is a recent topic of great interest in this area. *Spam/phishing email content filtering*, derived from the domains of information retrieval, pattern recognition and artificial intelligence, classifies malicious and legitimate messages based on the understanding and analysis of the email body and subject. In order to understand the content of emails, content filtering needs a large amount of dataset for training in order to perform categorization properly. Many researchers evaluate the performance of content filtering algorithms to assure the quality of different content filtering algorithms. Most of these evaluation studies for quality assurance purposes focus on comparing different content filtering algorithms, i.e. finding out the *classification performance* among different content filtering algorithms [2].

Research Rationale: In addition to the filtering algorithms, another factor which may have an impact on the performance evaluation results is the email *corpus.* Corpus (of Latin origin with plural: *corpora*), is a set of email samples, including both unsolicited and legitimate samples, which are used for text analysis in order to train and test the content filtering features. Several security organizations offer their own corpora that are selected from a variety of sources e.g. *honey pots, volunteers*, and so on. However, these *private* corpora are not naturally devised and ideally qualified for performance evaluation purposes because of the limitations on their topics, available formats and other. Therefore, some other research studies have also been conducted to discover and examine the reliability of the performance evaluation results; in these studies different *public* corpora are taken into use and research consideration [3].

Research Objective and Research Question: From the perspective of information retrieval, a good text classification feature is not only equipped with superior artificial intelligence, e.g. machine learning algorithms. It should also possess a strong capacity regarding (i) information processing and (ii) retrieval methodology, e.g. features' extraction and features' selection techniques. Unfortunately, not so many research outcomes and related studies are currently evidenced in this research area. Hayati and Potdar, for instance, evaluate the content filtering utilising text classifiers from a more general perspective [4], but an analysis of the text classifiers from the information retrieval point of view is absent. The paper authors' main research objective is to assist evaluating content filtering in information retrieval discipline, and for this they are searching for the answers to the following research question: *"a) how to design a reliable performance evaluation on spam/phishing content filtering, and b) which are the answers revealed from reviewing the existing research outcomes in this area?"*

Paper Structure and Researchers Motivation: In the following, we firstly explain the criteria of selecting important literature materials. Next, we present the literature review results from the performance experiments on content filtering, where the experiment design was prepared concerning (i) popular text categorization algorithms, (ii) pre-processing steps and (iii) corpora. After that, we analyse and relate the results to further needs and introduce the research focus on the critical challenges in the real world. Finally, we reflect on the design quality of existing content filtering performance experiments and offer some advice on how to design reliable performance experiments on spam/phishing content filtering.

This work was motivated from the fact that there are no, at least to our knowledge, any meta-evaluation studies that examine the quality, and reliability in particular, of performance evaluation experiments on spam/phishing content filtering. It is also part of our ongoing research work on constructing practical and usable software design quality criteria to be considered by practitioners and software developers for the design and implementation of anti-phishing/anti-spam technologies.

2.0 Research Methodology and Literature Selection

Literature reviews are the useful and valuable summary of a collection of representative references, which outline the contributions and limitations of existing studies in a certain research domain [2, 3, 5, 6, 7, 8]. In order to provide a high-quality literature review, we followed the guidelines set by Järvinen in [9]. In Järvinen' s article, multiple literature review methodologies are introduced. The nature of content filtering and information retrieval (with multiple control variables and multiple pre-processing steps) and the lack of documented similar performance studies, make the *lens-directed approach* to be most suitable for our research focus. Further, in this research approach, classification dimensions are selected from existing alternatives when a researcher has a preliminary idea on the research topic.

For a comprehensive literature review, we also comply with the four classification rules in this approach, which are: (i) research oriented grouping property selection; (ii) exhaustive analysis of the same hierarchical rank; (iii) pair-wise disjoint, and (iv) coincidence of various natural classifications of the same universe of discourse.

Research oriented grouping property selection emphasizes that the properties and characters selected should be useful and consistent to the research requirements. In our research, the selected grouping properties are the elements in performance evaluation experiments, i.e. test preparation, test sequence, test subjects, test objects. Exhaustive analysis of the same hierarchical rank and pair-wise disjointness guarantee the comprehensiveness and representativeness of each classification. The exhaustive analysis in our study is to find out as many as possible valuable designs of performance evaluation experiments so that to assure that the selected literature contents are representative enough. The coincidence of various natural classifications of the same discourse indicates that the nature of the reliable classification criteria on the same object should be persistent. Our research should conclude the way(s) to design a performance evaluation experiment of email content filtering methods. Hence, it must comply with the nature of experiment design.

2.1 An Analysis of Preliminary Findings

After a preliminary analysis of existing research on evaluation experiments of content filtering [4], the researchers found that various variables and performance measurements were prepared before the experiments. These variables, such as *feature size, corpora for training, corpora for testing* and *filtering algorithms*, were the main issues when the experiments were designed. Feature size is to define the number of words, terms or tokens appearing in each email. Corpora for training is a set of email samples to train content filters. Differently, corpora for testing is a collection of email samples to test content filters. *Performance measurements* include *cost-sensitivity, precision, recall, accuracy rate*, and *error rate*. A detailed commentary on these variables and metrics is given in the following sections. Similar performance measurements are widely used in almost every research paper that analyses the performance of different spam/phishing content filtering algorithms. Therefore, only one dimension is selected for classification, which is the key variable in content filtering.

We believe that, the *key variables* in performance evaluation experiment design are *text categorization algorithms*, *pre-processing steps* and *corpora*. The popular text categorization algorithms with their performance compared and examined were:

1) Naive Bayesian

2) Support Vector Machine

3) Memory-based filtering

4) Maximum Entropy Model

These algorithms are used to calculate probabilities for an incoming email being a spam or phishing message.

In order to define the likelihood of an email being malicious, the algorithms actually calculate the probability of one set of features in an email; the features appear in known spam letters. Therefore, it is a critical part to manipulate or extract the proper features from the emails. The performance of these pre-processing steps to capture ideal features is also one vital metric. In our preliminary analysis, there are some academic materials [1, 9] focusing on the performance evaluation of these pre-processing steps in text categorization, which are good references for performance experiment design.

2.2 Performance's Reliability and Other Considerations

The quality of corpora also determines the reliability of the performance experiments on filters. First of all, the collection methodology of these corpora must be examined. Second, it is important to analyze the corpora both linguistically and semantically in order to discover the filtering performance in a wide language spectrum. Moreover, the reliability of corpora from the statistical point of view assists in calibrating the filtering error rate regarding errors in labelling corpora. A long time research in this classification area revealed rich materials available in academy and industry. Even though we followed our literature selection methodology for more value and relevance, the searches yielded plenty of reported results in papers related to our topics.

To eliminate the redundancy and give more focus to our literature review, it was needed to restrict our attention to a limited number of literature findings. The further selection of these findings was refined according to their publication date and relevance to our own research interests [11]. Hence, we firstly chose a classic and highly cited research paper [12] as a starting point of performance evaluation on content filtering. After that, we continued our research with this paper acting as a benchmark. When the filtering algorithms, pre-processing, corpora treatments in the performance experiments were different, they were collected. The key words to look for literature review material were extracted after considering a preliminary analysis in the first paper, marked as number [12] in the list of references. For example, in paper [12] the performance of the Bayesian algorithm was conducted; so we searched for similar algorithms and their performance evaluation experiments of other text filtering algorithms in spam prevention area. In so doing, we restricted the search and finally retrieved rich and relevant literature findings discussing the performance of artificial intelligence based anti-spam technologies. The literature findings searched and collected from the popular academic and public search engines are listed in Table 1.

	Citeseer	**Google**	**ACM**
Performance Evaluation on Algorithms	1483	9840	756
Studies on Pre-processing Steps	64428	6710	223
Studies on Corpora	13894	9840	163

Table 1: Search results of spam/phishing content filtering from selected web engines

For the literature reviews conducted, we firstly found out the mostly cited papers, and then manually collected the ones whose topics were closest to our research domain.

3.0 Performance Experiments on Content Filtering: Analysis of their Strengths, Weaknesses and Limitations

A brief description of every content filtering performance experiment that is related to our research is provided in this section. A critical analysis also takes place at the same time on the most relevant to our research question (see Introduction part) issues. Thus sections 3.1-3.4 outline the contribution, weak points and limitations of the performance evaluation experiments on content filtering.

3.1 Performance Experiment on Naive Bayesian Filtering Approach

Sahami and his colleagues [12] are one of the first researchers introducing the Bayesian approach to filtering junk emails [12]. In the research of Sahami et al. [12], the Bayesian algorithm and its application conditions are defined. For junk emails classification, it is assumed that the terms in the junk emails are independent. With this assumption, a document classification problem is reduced down to the problem of calculating how likely an email is classified as spam when a set of terms appear in the document. To illustrate the effectiveness of their email classification solution, the authors also devised three experiments in junk email detection. To eliminate the size of features, the *mutual information (MI)* metric was applied. Mutual information is one metric to measure the dependence degree between the feature attributes and their category [13].

After analysis and comparison of MI results, the authors limited the number of feature size to 500, which is the greatest value when building a reliable classifier. For the first experiment, Sahami and his colleagues used three types of corpora for training; these were manipulated and represented with word-based, phrasal and non-textual domain-specific features. With 99.9% as a classification threshold, the first experiment showed *high precision* and *recall rate* on both junk and legitimate classes. The classifier with all three types of corpora trained gave best results.

In the second experiment, the sub-classes of junk emails were examined. The results of the experiment showed that both precision and recall rates got worse. To evaluate the filter's performance in a practical situation, e.g. considering that a filter should be able to classify the true junk emails when they were read once and discarded, the authors tested the filter with the messages collected during the week following the time from which the training data were collected. The test results demonstrated the feasibility to deploy the filter for commercial email applications.

3.2 Performance Experiment on Comparing Naive Bayesian Filtering with Memory-based Approach

The research by Sahami et al. [12] confirmed the feasibility of the Bayesian filter deployment. However, the benefits of adopting the Bayesian method rather than other algorithms were not illustrated until Androutsopoulos et al. [5] conducted a comparative study between a Naive Bayesian and a memory-based approach. In a memory-based filtering spam emails are classified based on all the training instances stored in the memory. The memory-based algorithm used in this experiment is *k*-nearest-neighbour (*k*-nn) algorithm implemented in TiMBL [14]. In *k*-nn methodology the aim is to find the *k* number of training instances that are closest to the email instance to be classified. In the paper, the selected *k* values were 1, 2 and 10. That means, e.g. when *k* is given the value 1, only one classified instance is selected and used to calculate the distances from this classified instance and the new instance to be classified. When this new instance is closer to the one spam instance, it is classified as spam. Otherwise it is a legitimate email. One public and one private corpus for test and training were taken into use.

Androutsopoulos and his colleagues also applied MI to find certain amount of feature attributes (i.e. words) with the highest MI-score, to limit the feature size. The selected number of feature attributes varies, and the authors determined the feature size according to the *misclassification cost*. Misclassification cost (**λ**) indicates the cost of *false positive* and *false negative* in the experiment. Inhere, false positive means that legitimate messages are misjudged as spam and false negative means that spam messages are misclassified as legitimate.

In order to compare the effectiveness between Bayesian and memory-based algorithm, Androutsopoulos et al. introduced the *total cost ratio* metric for measuring the performance of content filtering algorithms. The total cost ratio (TCR) metric compares the cost between that spending on manually deleting spam messages without any spam filter and the time to process false positive and false negative emails, which can be simply expressed as follows:

$$TCR = \frac{N_{spam}}{\lambda \cdot n_{legit->spam} + n_{spam->legit}}$$

Androutsopoulos et al. found that feature size and the misclassification cost all together effect on the performance of Bayesian content filtering in non-linear way. In the experiment, the result shows that the Bayesian content filtering outperforms when the feature size and the misclassification cost factor are set as in the Table 2, in section 4.

3.3 Performance Experiment on Multiple Machine Learning Content Filtering Algorithm

In another research study Androutsopoulos et al. [6] found that pre-processing steps like *stop-list* and *lemmatization* do not significantly improve the performance of Naive Bayesian algorithm when the MI feature selection method is employed. Stop-list is a set of removed words when processing a document, e.g. the article words *a, an* and *the.* Lemmatization is the act to uniform the words into their normal form. For example, after lemmatization, 'better' turns to 'good'. However, these research results may not always be valid, especially when multiple classification methods, feature selection methods and corpora are taken into account.

Therefore, Zhang et al. [2] conducted a study to discover how the feature space sizes affect the filtering accuracy, and how the same filtering algorithm acts on the languages other than English. In their solution, document frequency, MI and χ^2 (CHI) tests were used to reduce the feature size. Document frequency is used to compute the number of sample documents containing the term. Like MI, CHI test result means the relevance degree between the selected feature and the class. The key performance measures in use were the same as Androutsopoulos' research. The innovative contribution of Androutsopoulos et al. was to use these three types of feature selection methods and four public corpora to evaluate five different algorithms.

Besides the Naive Bayesian and memory-based algorithms, three more algorithms were in use: The *Maximum Entropy*, the *Support Vector Machine* and the *Boosting*. The Maximum Entropy Model is used to build a model to decide on the probability distribution of a class which has the *maximum entropy*. With the probability distribution of the class, an email can be classified as spam or legitimate according to the probability distribution. The Support Vector Machine finds the *hyper-plane* between two classes so that the maximum distance from the hyper-plane to the nearest instance of each class. Boosting is used to build a strong rule which consists of multiple weak rules. By training, these weak rules can be adjusted with minimum error.

Zhang et al. [2] run two experiments: one was on the impact of feature selection method, and the other was a cross-classifier evaluation with corpora in *multiple languages*. Even though the second experiment illustrated that each classifier performed with the similar classification capability on multiple languages, the feature selection methods on different classification algorithms are more valuable compared to the experiment by Yang and Pedersen [1, 9], who also examined the effects of several feature selection methods on classification algorithms, but with a limited number of classification algorithms [9]. In Zhang's experiment, five advanced classification algorithms were discussed. The result shows that MI and CHI metrics are more effective than *document frequency* (DF) when the feature set size is small. Interestingly, using features from email headers alone results in better performance than using features from email message body.

Even though Zhang et al. [2] contributed to investigating how the different feature selection methods affect the performance of various classification algorithms, their efforts on the corpora were limited. For instance, while Zhang et al. used four public corpora, their reliability was not examined. To assure the quality of public corpora for training and testing, Cormack and Kolcz conducted an evaluation on spam filters with imprecise *ground truth* [3]. Ground truth refers to how reliably the training corpora and the evaluation corpora are used in performance evaluation experiments. Instead of focusing on spam filters, the authors mainly discussed how well spam filters perform when considering the labelling errors in training corpora.

In this experiment, the researchers used two public corpora: TREC 2005 [15] and CEAS 2008 [16]. Both of them are well known conferences and platforms for providing reliable resources on text and electronic message processing testing and evaluation. *User adjudication*, i.e. users' own personal judgements on incoming emails, is introduced to simulate users' feedback when receiving an email [7]. The performance measures in use are: (i) receiver operating characteristics; (ii) the area under the curve, and (iii) logistic average misclassification rate that are calculated based on false positive and false negative classification results. By running seventeen 17 spam filters on the corpora with TREC labels and SpamOrHam labels, TREC labels are more accurate. In their paper [2], the authors also introduced automatic and semi-automatic methods to reduce the labelling error in performance evaluation of content filtering.

3.4 More observations and challenges for reliable evaluation experiments on content filtering performance

It is also worth mentioning that all the previously exposed and commented research studies were conducted by the researchers in the laboratory, and they were, in a sense, limited like all laboratory experiments, where variables can be controlled. Thus, none of the previous laboratory experiments took into account the real, live status of spam/phishing emails and their prevention. Therefore, it is also very important to collect data from the real world, with field experiments, where variables cannot always be controlled. Field experiments in this case are of great significance for (i) keeping track with the evolution of the spam/phishing technology and (ii) improving the performance of content filtering effectively. These are the reasons for which Fawcett kept surveillance on the status of spam and its challenges [17]. He listed four major challenges, which are listed and briefly explained next:

1) skewed and drifting class distributions;
2) error cost;
3) disjunctive and changing target concept;
4) intelligent adversaries.

1) Skewed and drifting class distributions concern the reliability of collected spam samples. According to the reports from SpamCop' s website [18], both spam messages and legitimate messages are collected and used in the research. Because of the huge difference in the spam class monthly, it is not reliable to compare the performance evaluation results among the spam filters whose spam samples are collected within different periods.

2) Error cost is concerned about the impact of misclassification, i.e. false positives and false negatives.

3) Disjunctive and changing target is a huge challenge focusing on the problem created by the epidemics topics of spam messages and their variations in different time periods. For example, in January most of the spam emails deal with donation requests, while in February the favourite topics may be gambling games invitations.

4) Intelligent adversaries should attract enough attention. This is because these adversaries keep tracking of the new prevention technologies so that they can always successfully bypass various filters.

The final section of our work refers to the above and other challenges and suggests how these should be considered in the future research agendas on anti-spam and anti-phishing software technologies.

4.0 Conclusions and Future work

Various content filtering performance experiments have been conducted in the subject area. These studies were devised and implemented based on the information retrieval methodologies, focusing especially on key variables such as filtering algorithms, pre-processing steps, corpora and performance measures.

Table 2 illustrates the values of feature size and misclassification cost factor as well as the feature selection method for filtering algorithms when perform best. As can be seen and concluded from Table 2, the Naive Bayesian algorithm outperforms.

Content Filtering Algorithms	**Feature Size**	**Misclassification cost factor**	**Feature Selection Method**
Bayesian	100	9	IG
	300	999	IG
SVM	10000	9	IG
	500	999	IG
Adaboost	9000	9	IG
	500	999	IG
Maximum Entropy	8000	9	CHI
	6000	999	CHI

Table 2. The values of feature size and misclassification cost factor for filtering algorithms when perform best

Feature selection methodology, feature size, and misclassification cost factor are the three key variables in the design of content filtering experiments. In general, CHI and IG are better choices than DF, although feature selection methods are sensitive to different classifiers. Table 2 lists the recommended values of these three variables that classifiers can reach best performance of their own. For example, the performance of Naive Bayesian classifier touches the best, when the collected features reaches 300, information gain is used to select features and the misclassification cost is set to 999.

In spite of a large amount of studies in performance evaluation on content filtering, more research is needed when new attacks occur and prevention methods take place. First, the growing number of image spam messages raises the requirements to understand the performance of image spam filters [4]. Moreover, new machine learning methodologies, like e.g. ontology engineering and semantic web technologies, are introduced to prevent spam/phishing messages [18]. This also demands updated performance evaluation design, which is able to identify and control the variables in the new environment.

As the essential element in performance evaluation, corpora should have been attracted more attention. The critical limitation in current “bag of words” treatment is that the email context has been ignored. However, according to the report from [17], one of the trends of spam/phishing attacks is that the topics vary significantly in periods. So far, there are no laboratory or field experiments on how the skewed corpora topics distribution impact the effectiveness of diverse content filtering, even though some contributions have been made on investigating junk emails linguistically [19].

Concerning the fourth challenge listed in Fawcett’s research in section 3.4, that is *intelligent adversaries,* Webb and his colleagues [8] made a valuable exploration to evaluate the performance of several classification algorithms by feeding with camouflaged corpus [8]. The content of this camouflaged corpus consisted of information from both spam and legitimate messages. In this way, the researchers discovered and suggested to train the algorithms with the original training set and get trained to treat all camouflaged messages as spam. We also support that it will be a valuable attempt to scrutinize the noise impact on the classification algorithms, while the evaluation methodology may need further refinement.

Following our literature review findings, we can support that the design of performance evaluation experiments is prone to be similar, especially for the machine-learning-based content filtering algorithms. The design of these experiments also has the same limitations and negligence. The most critical blank is the lack of efforts on corpora. Firstly, different languages may impact the performance of classifiers [2]. Secondly, the noise, e.g. camouflaged messages in [8], could significantly impact the performance of classifiers. Therefore, we strongly feel the urgent need of the well-grounded research on corpora design, especially when introducing noises into corpora.

We conducted this literature review in order to discover the strengths, weaknesses and limitations of many performance evaluation experiments of email content filtering methods. This study constitutes a natural part of our ongoing research project [21, 22, 23, 24, 25] on establishing software design quality criteria for anti-phishing/anti-spam software technologies, hoping that these will be invaluable for software application users and software developers.

Inevitably, the spam/phishing content filters will continue utilising and applying artificial intelligence and machine learning capabilities. In order to evaluate the performance of these various technologies, it is necessary to establish unbiased ways. This means that the adopted evaluation method should not focus merely on the feature size, misclassification cost, and filtering algorithms. It is equally important to also focus on the diversity of the samples, e.g. how spam filters perform when noises are added to email samples, or how the different sizes and types of attachments affect the processing accuracy and speed.

Our future research work considers the latter, among other issues, for defining and formalising quality criteria for software design of anti-phishing and anti-spam technologies. Last but not least, more field experiments than laboratory experiments are required in order to evaluate the performance of anti-spam/anti-phishing technologies and capture the real challenges and security problems that occur in online communication.

5.0 References

[1] Microsoft, Antispam Technology, http://www.microsoft.com/mscorp/safety/technologies/antispam/default.mspx

[2] Zhang L, Zhu J, Yao T, An evaluation of statistical spam filtering techniques, ACM Transactions on Asian Language Information Processing (TALIP), 3(4), 243-269.

[3] Cormac G, Kolcz A, Spam filter evaluation with imprecise ground truth, Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pp604-611, Boston, Massachusetts, USA, 2009

[4] Hayati P, Potdar V, Evaluation of spam detection and prevention frameworks for email and image spam - A state of art, Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services, pp520-527, Linz, Austria, 2008

[5] Androutsopoulos I, Paliouras G, Karkaletsis V, Sakkis G, Spyropoulos C D, Stamatopoulos P, Learning to Filter Spam E-Mail: A comparison of a naive Bayesian and a memory-based approach, Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000), pp1-13, Lyon, France, 2000

[6] Androutsopoulos I, Koutsias J, Chandrinos KV, Spyropoulos CD , "An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with encrypted personal e-mail messages, Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000), pp160-167, Athens, Greece, 2000

[7] Cormack G V, Lynam T R, Online supervised spam filter evaluation, ACM Transactions on Information Systems, 25(3), 2007

[8] Webb S, Chitti S, Pu C, An experimental evaluation of spam filter performance and robustness against attack, International conference on Collaborative Computing: Networking, Applications and worksharing, 2005. Retrieved November 11, 2011, from Georgia Institute of Technology: http://www.cc.gatech.edu/~webb/Papers/Webb_CollaborateCom_2005.pdf

[9] Järvinen P , On developing and evaluating of the literature review, Technical Reports. 2008 Retrieved December 25th, 2010, from University of Tampere: www.cs.uta.fi/reports/dsarja/D-2008-10.pdf

[10] Yang Y, Pedersen J O, A comparative study on feature selection in text categorization, ICML '97 Proceedings of the Fourteenth International Conference on Machine Learning, pp412-420, San Francisco, USA, 1997

[11] Geist M, Enhancing Home Computer User Information Security: Factors to Consider in the Design of Anti-phishing Applications, Master Thesis, University of Oregon. Retrieved December 23th, 2010, from University of Oregon: https://scholarsbank.uoregon.edu/xmlui/handle/1794/7653

[12] Sahami M., Dumais S., Heckerman D., Horvitz E.(1998), A Bayesian Approach to Filtering Junk E-Mail, Learning for Text Categorization: Papers from the 1998 Workshop. AAAI Technical Report WS-98-05.

[13] Cover T.M., Thomas J. A., Elements of Information Theory. Wiley. 1991

[14] Tilburg Memory-Based Learner, 2010 , ilk.uvt.nl/timbl

[15] TREC 2005 Corpus, Retrieved February 16 2011, http://plg.uwaterloo.ca/~gvcormac/treccorpus/,

[16] CEAS 2008 Corpus, Retrieved February 16 2011, http://www.ceas.cc/2008/

[17] Fawcett T. (2003), "In vivo" spam filtering: A challenge problem for data mining, KDD Explorations, 5(2) http://home.comcast.net/~tom.fawcett/public_html/papers/spam-KDDexp.pdf

[18] SpamCop, SpamCop web site. Retrieved December 30, 2010, http http://www.spamcop.net/

[19] Youn S, McLeod D, Efficient spam email filtering using adaptive ontology , proceedings of International Conference on Information Technology, pp249-254, Las Vegas, USA, 2007

[20] Orasan C., Krishnamurthy R., A corpus-based investigation of junk emails, Language Resources and Evaluation Conference, Las Palmas, Spain, 2002

[21] Li, L., Helenius, M., Berki, E. Phishing-Resistant Information Systems: Security Handling with Misuse Cases Design. Conference Proceedings of Berki, E., Nummenmaa, J., Sunley, I., Ross, M. & Staples, G. (Eds) *Software Quality in the Knowledge Society. SQM 2007.* pp. 389-404, Tampere.

[22] Jäkälä, M. & Berki, E. Exploring the Principles of Individual and Group Identity in Virtual Communities. Commers, P., Isaias, P. & Baptista Nunes, M. (Eds) Proceedings of the *1st IADIS Conference on Web-based Communities*. Mar 24-26 2004, Lisbon. Pp 19-26. International Association for the Development of Information Society (IADIS): Lisbon

[23] Berki, E. (2001). *Establishing a Scientific Discipline for Capturing the Entropy of Systems Process Models. CDM-FILTERS. A Computational and Dynamic Metamodel as a Flexible and Integrated Language for Testing, Expression and Re-engineering of Systems.* Ph.D. Thesis. Faculty of Science, Computing and Engineering. University of North London.

[24] Li, L. and Helenius, M.: 'Usability Evaluation of Anti-phishing Toolbars', Jounal of Computer Virology 2007, (3), pp 163-184

[25]. Berki, E. & Jäkälä, M. 2009. Cyber-Identities and Social Life in Cyberspace. Hatzipanagos, S. & Warburton, S. (Eds) *Social Software and Developing Community Ontologies* (London: Information Science Reference, an imprint of IGI Global). Pp.28-40.

Study 3

Li L., Helenius M., Berki E. (2011). How and Why Phishing and Spam Messages Disturb Us? *Proceedings of Bradley G. (Ed) IADIS International Conference ICT, Society and Human Beings 2011*, Rome, 20-26 July, 2011, pp.239–244.

HOW AND WHY PHISHING AND SPAM MESSAGES DISTURB US?

Linfeng Li[1], Marko Helenius[2] and Eleni Berki[1]
[1]*University of Tampere, Finland*
[2]*Tampere University of Technology, Finland*

ABSTRACT

Further than privacy and security, social engineering techniques employed by sophisticated spammers and phishers have a negative impact on personal and professional lives. In this paper, the authors comment on how severe and annoying malicious messages are. In so doing, we encouraged several spam/phishing email receivers from different countries to forward their fraudulent multilingual messages of various contents. We analysed them and subsequently conducted email interviews with the receivers. Additionally, we compared and contrasted the information we obtained with the internationally reported phishing scams from the Anti-Phishing Working Group. The paper discusses (i) the impact and feelings generated from fraudulent emails on people who receive them, (ii) the ways people interpret their content, and (iii) how they react on it. We made a snapshot of phishing email with our assumption being that spam/phishing resembles each other; so taking a snapshot we made a good estimate of various phishing email techniques. Summarising, we put the emphasis on further searching and analysing feelings and emotions of humans. These are important considerations while addressing human-centred quality design features of anti-phishing software technology.

KEYWORDS

Social engineering; phishing/spam email; security; human-centred quality design; software user psychology

1. INTRODUCTION

Email technology has been established as one of the most important online communication tools for organizations and individuals. Similar to other more traditional communication tools, e.g. land line telephone, email lacks the sufficient functions and information to identify the two parties at the two ends of a conversation. The vulnerability in the verification capability of emails leaves a gap for technology misuse.

Background: Spam messages are messages full of commercial and unwanted information to be delivered to the email accounts. The main purpose of these messages is usually not to control the victims' computers. Instead, it is to carry out promotional activities. Phishing attacks are more aggressive and risky, baiting potential victims with fraudulent web services in order to finally be seduced to provide their personal information, especially the confidential personal information like usernames, passwords, bank account numbers and other. The phishing attacks are usually carried out and delivered through emails. The worst situation is when a phishing attack is launched together with some other subterfuge technologies, e.g. viruses and Trojan horses. These technologies can take over users' systems, record and subsequently transmit users' activities and personal data and information (E-Mail Security, 2011).

To prevent spam and phishing email, many researchers have resorted to intelligent ways (Androutsopoulos et al., 2000; Zhang et al., 2004; 2007; Webb et al 2005; Wu et al. 2006) to efficiently detect and prevent these messages. Other researchers keep looking for solutions in optimizing phishing preventions from the usability perspective (Dhamija et al., 2006; Li and Helenius, 2007). Among these usability studies, it is worth-mentioning that Dhamija et al. conducted a study to find out *why* phishing works. This research mainly focused on how end-users interpret phishing web pages and how they react on spoofing pages. Other researchers prompt to understand how spam/phishing email evokes individual feelings and emotions and how usability and other quality design features could be enhanced (Li et al., 2007; 2011) through the analyses of personal email experiences and reactions to malicious email contents. A summary of design principles for motivational affordance in (Zhang, 2011) indicates that most relevant to ICT use are psychological, social, cognitive, and emotional sources of motivation. Motivational affordances are fundamental reasons for ICT design and use.

Research Question and Research Methodology: Even though many researchers contributed with a lot of research studies to the previously described research fields, it is still needed to know the nature, content and consequences of these malicious messages. In particular, it is important to know how these are considered by ordinary email users. In this paper, we try to find the answer to the research question: *"how severe and annoying these malicious messages are, and why?"* For answering this question, we invited various email users from different countries to forward their spam messages for our analysis. We subsequently contacted the *same* email receivers for a further email interview to reveal different personal reactions and feelings. We made a snapshot of phishing email with our assumption being that spam/phishing messages resemble each other; so by taking a snapshot we made a good estimate of various phishing email techniques. One of our paper's novelties is that we have different recipients from different countries and emails in many different languages. The email samples were thoroughly analysed. Although the spam/phishing randomly chosen email receivers were of different nationalities and spoke different languages, the interviews were conducted only in English, which is easy for the participants and us to interpret.

In the following, we firstly review the phishing attacks' severity utilising recent information from the international Anti-Phishing Working Group reports 2008-2010. Afterwards, we present the characteristics of the spam/phishing samples we received. Subsequently, we expose our interview questions to some of the receivers of these emails and based on the answers we discuss how these email users understand spam and phishing messages. Concluding, we provide our insights on the dangers of spam/phishing messages and emphasise the need to examine the impact of unsolicited emails to humans. A deep analysis is needed to understand email users psychology in order to incorporate their views and preferences with security considerations in the quality features for anti-spam/-phishing software technologies.

2. THE PHISHING TREND FROM THE ANTI-PHISHING WORKING GROUP VIEWPOINT

In this section, we briefly introduce the existing and popular phishing preventions providing also the reader with some primary analysis based on our ongoing research work so that the reader understands better (i) how and why phishing works well and (ii) why phishing preventions cannot work perfectly.

Research Context: According to the most recent reports from APWG, there exist a big number of *reported* scams. Next, we are reviewing these scams in a historical order. To limit the research scope and efforts, we start to review these reports from the one published in January 2008. Time-wise, a two-year period is long enough to analyze the trend and severity of the phishing attacks. Another reason for our choice to deal with this period is the international financial crisis that burst out in 2008. From the social sciences and economics perspective, it is of great research interest to find out if, in periods of socio-economical crises (i) the phishing attacks phenomena increase/decrease and (ii) phishing attacks email contents are related to the apparent needs of phishers and spammers. In fact, during this time period, a rise of more and more sophisticated techniques used by phishers has been observed; such are, for instance, *key logger and screen logger* which resulted in more extensive counter-measures and research efforts for anti-phishing and anti-spam technologies in academy and industry.

APWG Report of 2008: 90% of the reported phishing sites are monetary organizations. Notably, statistical data on the techniques used by phishers were not highlighted in the main reports. The issue, however, was discussed carefully in a project called *Crimeware*. In this project there were two phishing-based Trojan attacks introduced, namely *key loggers* and *re-directors* (Anti-phishing Scams, 2011). Key loggers try to steal users' accounts' details and passwords by recording, in a sneaking way, the key strokes in the corresponding text input fields and password input fields on web pages. The statistics show that this phishing technique was utilized at a relatively high level. Further, there are 10% of the reported phishing websites that were hosting password stealing malicious code. Redirectors, working together with key loggers, try to redirect users' requests to pre-designed malicious locations. Unfortunately, no statistics on redirectors were available. The number of reported phishing sites remained high and the malicious code unique applications to steal users' personal information were widely and increasingly spread out globally. Notably, the number of password stealing malicious code URLs grew dramatically during December 2008 and reached 31 173, and this happened during the first Christmas holidays after the outbreak of the financial crisis.

APWG Report of 2009: Besides the growing number of reported phishing sites in the reports for 2009, new *rogue anti-malware* programs appeared to receive more attention and emphasis on their content and purpose. Rogue anti-malware programs usually prompt a number of false negative alerts to persuade users to buy these or other rogue programs. They are rogue also because they may compromise the system while it is hard to uninstall them, and at the same time they claim to be anti-malware software. The number of detected and reported rogue anti-malware programs reached the peak twice, in June and December in 2009. More interestingly, according to the statistics from Panda Labs (Anti-phishing Scams, 2011), about one out of four computers were infected by the malicious codes which may help result in more successful phishing scams.

From the meta-analysis of the two years' APWG reports, we can conclude the following two interesting findings. First, money-related organizations are the ideal targets for phishers, especially in hard times of socio-economical crises. Second, phishing techniques have become increasingly sophisticated at every step in simple, deceitful, and apparently trustworthy scams. Sending emails with malicious attachments, or redirecting users to run or install malicious codes is a few of the very seriously considered fraudulent phishing attempts with disastrous consequences.

3. CONTENT ANALYSIS AND CLASSIFICATION OF THE REPORTED SPAM/PHISHING EMAILS

During the years 2009-2010 we asked different email account owners from all over the world to forward to us each email they received and they thought was Spam or Phishing email. We collected 178 malicious emails from the email users. The content of the emails was written in the following languages: English, Chinese Mandarin, Chinese Cantonese, Finnish, Greek, Spanish, Italian, German. The subjects/email receivers were men and women of different ages, nationalities and occupations from China, Germany, Greece, United Kingdom and Finland. We structured and classified the basic information of the email contents. A detailed description of these emails is exposed below. Many share similarities while the persistence and repeatability of the content is notable.

There are many quite similar phishing emails resembling each other. In fact, it seems easy to broadly categorize the mails as follows: *1) Nigerian fraud* (lottery winning, money transfer, dating or contract); 2) *phishing mule recruitment; 3) user account stealth; 4) credit card information stealth; 5) malware installation; 6) online casino and 7) product sale* (electronics, watches, general e-shopping, pharmacy, Viagra, degree certificates and sensitive database records).

The phishing attempts in emails are, in most cases, obvious when you know what to look for. This knowledge is a considerable matter also for ordinary users since by learning the characteristics and following simple rules it seems easy to avoid phishing emails (E-Mail Security, 2011; Li and Helenius, 2007). However, in our email collection there were a couple of cases where it was hard to say if it was legit bulk mail or not. For example, there was an email from a legit online travel agency, but there were complaints about bad customer service and non-refund of money on Internet discussion lists.

An appealing trick was to use a friend's email address, but this was used only once. Appealing account reset email may be challenging, too, especially when it is well written and the source email is forged so to look like coming from a legit source. Similarly, in one Nigerian Fraud, a Microsoft's email address was faked; the same also occurred in one Google address. Additionally, in some Nigerian fraud emails authentic news' links were used to convince on the story-telling. Most of the related web addresses in phishing emails did not seem to work as originally planned. Obviously, illegal sites seemed to have been taken down. There seemed to be also some legal bulk email that the recipients considered as spam or phishing. Of course, it may still be spam for them even if it is legal; or the sender did not have a legit reason to send the email.

In spam the sender always appears to be different from the contact information. However, this is true also in the case of legal bulk mail and, thus, phishing mail cannot always be recognized by this characteristic. In phishing and spam mails there is typically no opt-out option. However, in one phishing mule recruitment seemed to be a faked opt-out option. In legal bulk mail there was always a possibility to opt-out.

Often there is a language barrier which prevents successful phishing. For instance, among the specimens there were two phishing attempts in Finnish, and other in Greek or Spanish or Italian, but they were so badly written that it was tiring to read and hard to follow their content. Only by knowing the expected fraudulent content it was possible to guess the intentions of the particular mail and its senders. The above observations raised to us the interest in finding out more on the recipients' feelings and reactions while reading these emails. So, we further asked the recipients some simple questions. Next, there is a tabular representation and analysis of the replies that six of the email receivers sent to us.

4. INTERVIEWS OF THE EMAIL USERS

Research methodology issues: In our data collection, we used the unstructured interview without scales (Psychology, 2011). First, we maintain that it is valuable to become acquainted with the profiles of these people and know a little more especially about the way email is used and other related personal information. This information proved to be valuable to us while we further analyzed and attempted to classify the contents of emails the research participants received.

4.1 Choosing Specific Interview Questions

Some of the personal information could be measured in scales, but some measurements are not scaled so that the details and integrity of the interview results can be kept. For example, instead of scaling participants' feelings, we asked them to give more subjective answers to describe the troubles spam messages bring to them. Often these descriptive answers contain more information for the subject researchers. Further, few questions were given to reduce redundancies and duplications. These questions are:

Q1. How long time have you used computers?
Q2. How often do you read email?
Q3. How much spam do you receive?
Q4. What type of messages are spam for you? (please give examples)
Q5. How spam affects you?
Q6. What type of spam is the most annoying?

The first three questions were designed to collect personal information related to email usage and the last three questions were designed to find out the participants' different attitudes on understanding spam and its severity. Below we present and analyze the answers provided by the participants interviewed.

4.2 Interview Results and Analysis

Six out of seven participants returned the answers to the questions. According to the interview results, it is believed that these six participants had sufficient experiences in using computers and email (Q1 and Q2). Comparably, all six participants received more than 5 spam messages per week (Q3). Astonishingly, the maximum number of spam one participant received in one day reached 25. When the participants were asked to give practical examples on spam (Q4), all of them asserted that various advertisements from unknown individuals and organizations are most likely spam. Another characteristic among these participants was that the messages with the topics they were/are not interested in were/are all spam.

Three of the respondents (participant 1, 3 and 4) gave the examples that people can easily identify their categories from subject and content fields of emails. Participant 2 and 5 presented one similar special example like *"From time to time people have different interests and hence visit various web pages, register in and subscribe to different web site services and e-mailing lists. When, however, they change interests, the emails from previously subscribed web sites or mailing lists are most likely to be considered as spam."*

When they were asked on the spam mail's influence (Q5), almost all participants were concerned with the time-consuming and malware infection risk of mistreating spam messages. Participant 2 emphasized the superior performance of anti-spam features of Gmail. It is true that some spam detection and prevention mechanisms outperform, but it is also important to acknowledge that this participant uses two different email services, one for work and one for personal use. The separate use provides the participant with the opportunity to significantly lower the cost on processing spam messages during the work time.

Finally, the most annoying spam message types (Q6) for different participants vary, though they surprisingly share the same rule and similar thoughts, defining the most annoying type as what is not included in the users' interests. Notably, the frequency of spam messages is also an annoying issue. The responses of the participants in their exact wording are exposed in a tabular representation next.

Table 1. Responses of the spam email recipients

	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6
Question 1	>10 years	>10 years	20 years	41 years	13 years	25 years
Question 2	Every day	Every day	Several times a day	Every day	Every day	Every day
Question 3	2-5 per week	1-3 per day	Moderately much	10-25 per day	10 per week	5-6 per week
Question 4	Irrelevant Ads, from anonymous sender, fake sender. User's email account is mentioned in the message.	Ads, weird link, website promotion.	Obscure events and sales proposals that I never dream of participating.	Drugs sales, financial deals that try to entice you to give personal details	Erotic messages and from the web sites I registered before.	emails related to any commercial products adverts, lotteries, notifications of winning presents, conference/workshops adverts invitations, etc.
Question 5	Spend time to delete. Possible risk if it contains a virus, or leak my personal info.	Never, because Gmail almost filter 99% of spam email	Waste of time, attention, resources	Irritating, destructive, annoying and embarrassing, time-wasting to delete them	Waste my time and mislead me to click some link or download malware	disturbs my thinking, it takes considerable time, diverts my thinking and alerts my e-moves, that is I am more careful!
Question 6	Irrelevant Ads.	'Weird link' and 'Website Promotion' types of spam	Political at home, tasteless at work	Financial deals, fraud, pornographic	Frequently received messages, spam link included	The newer ones, because it takes time and lots of thinking to recognise more and more sophisticated spam emails.

5. CONCLUSIVE REMARKS AND FUTURE WORK

After comparing our available information with the APWG and received data (forwarded emails and answers to the unstructured interview) we understand that spam and phishing emails have almost become part of the everyday or at least weekly life of people. We received only one sample of spam/phishing specimens which contain virus or malware. Actually, a very small number of these specimens contain attachments. APWG, however, reported that many malicious codes have been hosted on the reported phishing websites. A concise summary of the comparative findings is provided in Table 2. Obviously, when spam messages become more sophisticated, they can bypass the spam filters and they can also be so confusing and deceptive that receivers may take more time to read carefully and identify the legitimacy of these spam messages. Beyond the various filters' performance and effectiveness, a good practice for email users is to use separate email services for different purposes, e.g. separating work emails from personal emails. In this way, it raises the accuracy rate of spam preventions, and psychologically also cuts the spam misclassification cost.

Table 2. Summarised findings from comparisons between feedback from real email users and reports from APWG.

	Feedbacks from real end users	Reports from APWG
Malware involvement	1 virus/malware sample was contained in the collected emails from six participants.	Many phishing websites host malware, but no statistics are available.
Suggestions on dealing with false positives	Separate email accounts for work and personal use.	Not available
Spam/Phishing cost	Disturbance, danger, time loss, productivity	Financial loss

Beyond productivity and time loss, people's security and privacy are being compromised every day when millions of spam/phishing emails arrive at millions of email users' destinations, endangering their online presence and threatening to steal and use other personal details.

Current anti-spam/anti-phishing technologies that include metrics' approaches, filters, and detection policies provide some satisfactory solution for existing information systems only. Future information systems cannot benefit from these because they will be more advanced technologically and the malicious email senders more knowledgeable and more sophisticated. Therefore, a drastic and influential approach towards email users' protection needs to emerge considering to combine three important issues: (i) software user psychology; (ii) human-centered software design quality criteria, and (iii) software/email exploiters' cognitive profiles. In so thinking, future information systems can be designed with generally verified and acceptable quality criteria, enhanced security and preventive maintenance in mind, and will also have to take into account software users' psychology and their usability needs.

6. LIMITATIONS

Though we carried out a comprehensive analysis on how and why spam and phishing emails annoy people in everyday life, our current work has some limitations. First we had only 6 respondents, which is too small number for statistical analysis. Also the email samples that we received do not demonstrate all phishing email techniques. However, our original aim was not to get statistically holding results. The research aim has been to provide an improved understanding of the problem domain and produce new information that would show the impact of the particular ICT on human beings. Statistical results can be obtained from various Internet monitoring sources, like APWG. Our aim was to understand the problem more deeply by exposing and classifying the content of fraudulent emails and their impact on people's every day lives. Future research plans to enrich this classification and impact by examining more email contents and email recipients.

ACKNOWLEDGEMENTS

Accomplishing this study would have been very difficult without the willingness of the invited participants, to contribute with their email samples and personal experiences. We received great help and herein we want to express many thanks to all of them.

REFERENCES

Androutsopoulos I, et al, 2000. Learning to Filter Spam E-Mail: A comparison of a naive Bayesian and a memory-based approach, *Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Lyon, France, pp. 1-13.

Anti-phishing Scams, 2011, http://www.antiphishingscams.com/phishing-prevention.html

Anti-phishing Working Group, 2008-2010. *Phishing Attack Trends Report.* Anti-phishing Working Group.

Dhamija R., et al, 2006. Why phishing works. *Proceedings of the SIGCHI conference on Human Factors in computing systems*. Quebec, Canada, pp. 581-590.

E-mail Security 2011, http://email-security.blogspot.com/2010/07/spam-spoofing-phishing-pharming.html

Li, L. and Helenius, M., 2007. Usability evaluation of anti-phishing toolbars. In *Journal of Computer Virology*, No. 3, pp. 163-184.

Li et al, 2007. Phishing-Resistant IS: Security Handling with Misuse Cases Design. Software Quality Management Conference Proceedings: *Software Quality in the Knowledge Society.* pp. 389-404, Tampere 29.7-2.8.2007. British Computer Society: Swindon.

Li et al, 2011. Design of Reliable Spam/Phishing Content Filtering Performance Evaluation. In the Conference Proceedings of Software Quality Management 2011. Loughborough. British Computer Society. 19-23 April 2011.

Psychology, 2011. Unstructured Interview, http://psychology.wikia.com/wiki/Unstructured_interview.

Webb S, et al, 2005. An experimental evaluation of spam filter performance and robustness against attack. *International conference on Collaborative Computing: Networking, Applications and Worksharing*, San Jose, USA.

Wu, M., et al, 2006. Do security toolbars actually prevent phishing attacks? *Proceedings of the SIGCHI conference on Human Factors in computing systems*, Montréal, Canada, pp 601-610.

Zhang L. et al, 2004. An evaluation of statistical spam filtering techniques. *In ACM Transactions on Asian Language Information Processing*, Vol. 3, No. 4, pp. 243-269.

Zhang, P. 2011. Motivational Affordances: Fundamental Reasons for ICT Design and Use, *Communications of the ACM* http://melody.syr.edu/pzhang/publications/CACM_07_Zhang_Motivational_Affordances.pdf

Zhang, Y., et al, 2007. Phinding phish: evaluating anti-phishing tools. *Proceedings of the 14th Annual Network and Distributed System Security Symposium*, San Diego, USA, pp. 79-92.

Study 4

Li L., Helenius M. (2007). Usability Evaluation of Anti-phishing Toolbars, *Journal of Computer Virology* (3), pp.163–184.

J Comput Virol (2007) 3:163–184
DOI 10.1007/s11416-007-0050-4

EICAR 2007 BEST ACADEMIC PAPERS

Usability evaluation of anti-phishing toolbars

Linfeng Li · Marko Helenius

Received: 12 January 2007 / Revised: 7 February 2007 / Accepted: 26 March 2007 / Published online: 16 May 2007

Abstract Phishing is considered as one of the most serious threats for the Internet and e-commerce. Phishing attacks abuse trust with the help of deceptive e-mails, fraudulent web sites and malware. In order to prevent phishing attacks some organizations have implemented Internet browser toolbars for identifying deceptive activities. However, the levels of usability and user interfaces are varying. Some of the toolbars have obvious usability problems, which can affect the performance of these toolbars ultimately. For the sake of future improvement, usability evaluation is indispensable. We will discuss usability of five typical anti-phishing toolbars: built-in phishing prevention in the Internet Explorer 7.0, Google toolbar, Netcraft Anti-phishing toolbar and SpoofGuard. In addition, we included Internet Explorer plug-in we have developed, Anti-phishing IEPlug. Our hypothesis was that usability of anti-phishing toolbars, and as a consequence also security of the toolbars, could be improved. Indeed, according to the heuristic usability evaluation, a number of usability issues were found. In this article, we will describe the anti-phishing toolbars, we will discuss anti-phishing toolbar usability evaluation approach and we will present our findings. Finally, we will propose advices for improving usability of anti-phishing toolbars, including three key components of anti-phishing client side applications (main user interface, critical warnings and the help system). For example, we found that in the main user interface it is important to keep the user informed and organize settings accordingly to a proper usability design. In addition, all the critical warnings an anti-phishing toolbar shows should be well designed. Furthermore, we found that the help system should be built to assist users to learn about phishing prevention as well as how to identify fraud attempts by themselves. One result of our research is also a classification of anti-phishing toolbar applications.

1 Introduction

Phishing has become as one of the most serious network threats [5–7]. Similar to other malicious attacks, phishing can cause loss for both financial institutions and consumers. However, unlike most crackers, phishers gain benefits by accessing credential information, instead of system or network damage. Moreover, phishing damages the trust of e-commerce.

A devastating attack does not require any emerging techniques. According to the March 2006 report of the Anti-Phishing Working Group (APWG), the most frequently used artifices are deceptive e-mails or web pages, Trojan horses and key loggers. Moreover, more than 80% of fraudulent web domains contain ambiguous names, for example, some form of target name in the URL or only IP address without host name. These ambiguous domain names are hazardous for careless consumers.

So far, there are more than 10 academic research groups and 100 of governmental or commercial organizations contributing to phishing prevention [17] in both theoretical and practical areas. On the one hand some researchers try to find the way how phishing attacks are plotted [10], and investigate

Linfeng Li is a student at the University of Tampere, Finland. Marko Helenius is Assistant Professor at the Department of Computer Sciences, University of Tampere, Finland.

L. Li · M. Helenius (✉)
Department of Computer Sciences,
University of Tampere, Kanslerinrinne 1,
33014 Tampere, Finland
e-mail: cshema@cs.uta.fi

L. Li
e-mail: linfeng.li@uta.fi

how victims decide to trust phishing scams [3]. On the other hand, other researchers intend to delve how effective anti-phishing toolbars are from a technical perspective [19]. Wu et al. have made an usability evaluation about anti-phishing toolbars [18]. Their perspective concentrates on the human behavior while using toolbars. We concentrate on the design of anti-phishing toolbars. It seems that usability evaluation of anti-phishing applications is so far rarely researched domain.

Because of the careless usability security design, phishers can easily take advantage of poor usability design [9, p. 56]. In order to offer more reliable security, anti-phishing toolbars should be easier to use. Moreover, as end-users must be able to use the toolbars and make correct choices, usability evaluation of these toolbars is important [19].

Our research objective was to find out general usability design principles for anti-phishing client side applications. Such information may result in valuable information for improving usability and security of anti-phishing applications. Based on this motivation, we conducted the heuristic usability evaluation [14] of five toolbars. However, we must advice the reader that we are not making a comparison of the toolbars in this paper. An objective comparison would require a different approach and should concentrate on assessing phishing prevention capabilities.

In this paper, we will present our evaluation and discuss the issues found during the evaluation. In the following parts of this paper, we will first introduce the features and characteristics of these five toolbars in order to make readers aware of basic functionalities from a technical perspective. After that, we will present the heuristic evaluation methodology and the evaluation results we found. Based on the results, we will give advices for improving the toolbars' usability design. In conclusion, the usability evaluation is summarized, and the impact of weak usability performance of the toolbars is discussed.

2 Introduction to anti-phishing toolbars

We found that, there exist currently four basic types of toolbars, classified by their architecture and functionalities.

1. *Toolbars based on client–server architecture and anti-phishing prevention combined with other functionalities.* These types of toolbars need to communicate with their servers, in order to protect users from being spoofed. However, these kinds of toolbars are not tailored just for phishing prevention. Instead, there are other functionalities that are not related to anti-phishing. For example, Google's Safe Browsing functionality is only one of the toolbar's features. The other features include such as Enhanced Search Box, AutoFill, etc.
2. *Toolbars based on client–server architecture and designed only for phishing prevention.* These are also based on the client–server structure, but the functionality is only phishing prevention. Therefore, users can only find phishing related functionalities from their interfaces. For example, Netcraft toolbar is designed only for phishing prevention. Even though some of its functionalities are not directly associated with anti-phishing, these are designed to support identification of fraud web pages.
3. *Toolbars installed on the local computer and detecting fraud websites by user's browsing information.* Because of the lack of the server side, these kinds of toolbars have to use the browsing information or the browsing history for detection. This kind of data cannot be managed by toolbars themselves, but by web browsers. Therefore, it is required for users to configure the browsing records carefully.
4. *Toolbars installed on the local computer and detecting fraud websites.* Different from the previous type, these toolbars must use some other methods to identify spoofing websites, like a whitelist or general detection. Compared with the third type of toolbar, users may more freely customize their own preferences, e.g. authentic web sites.

In addition to these existing toolbar types, we observed that the classification can be developed further. For example, present techniques could be combined when developing toolbars further. The classification can be based on differences in architecture, detection method and identification mechanism. Some classification variables can be:

- Is the toolbar client–server based?
- What types of lists the toolbar uses for detection (blacklist, whitelist and/or graylist)?
- Does the toolbar use local history/cache information for phishing site detection?

In this evaluation, we chose four toolbars, in addition to our own. We are aware that there are also other toolbars. However, because of limited time and resources, we picked one typical toolbar of each existing type. We selected these toolbars according to two criteria. First, the selected toolbars must be common ones and downloadable from the Internet. Second, their capabilities to detect phishing sites should be satisfactory. Based on the Zhang et al. [19], we selected Google toolbar, Netcraft toolbar and SpoofGuard, which received highest scores in the evaluation. According to the comments of the reviewers' of our paper, the Phishing Filter in IE 7 is getting used by more and more people. Therefore we included it in the heuristic usability evaluation.

We also selected our anti-phishing IEPlug to be evaluated. One might be concerned that, because we have included our own toolbar in the evaluation, we are biased towards our own product. However, there are number of reasons for including our own toolbar. First of all, our meaning is not to place the toolbars in comparable order of goodness, but rather to find usability design principles in general level. Moreover, our aim was to improve the usability of our own application. Furthermore, anti-phishing IEPlug represents a different type of anti-phishing application. The application is not based on client–server structure and users manage their own whitelist explicitly. The warning method of the anti-Phishing IEPlug is not completely the same as with the other three toolbars. The application aims to improve user's security knowledge by actively showing the domains certificate authority (CA) information dialog. Finally, because our own application is open source application, we do not have any commercial interest.

We shall next present the toolbars included in the heuristic usability evaluation.

2.1 Google safe browsing

Google Safe Browsing (Fig. 1), is a part of the Google toolbar extension for the Firefox. It is able to alerts users based on a black list, when the web page visited is considered as a fraudulent one. There are two alternative choices for detecting fraud web pages. A user can select either "downloaded list of suspected sites" or "asking Google about each site I visit" [8]. When a user selects the first method, Firefox downloads or updates the blacklist each time, before a new Firefox window is opened. Whenever a user visits a page with Firefox, Google Safe Browsing will find out whether the page visited is in the blacklist stored locally. With the second detection method each address visited will be forwarded to a specified server maintained by Google. After the analysis, the server returns analysis results. When the page visited is considered as a deceptive one, the toolbar will stop the user's activity and give appropriate advice (e.g., stop visiting the web site, or ignore the warning). The user is also able to report mistakenly warned web pages.

2.2 Netcraft anti-phishing toolbar

The mechanism of the Netcraft anti-Phishing toolbar (Fig. 2) is similar to the Google's toolbar. It communicates with the Netcraft site's report database [13] and obtains the blacklist information. Moreover, the toolbar offers extra information concerning the page a user visits, including "RiskRating", "Since" (domains registration time), "Rank", and "Hosted server information". For example, when a user visits a page, he or she can be aware of the site's rank by following the link on the toolbar. However, this rank is based on the level of popularity, rather than security criteria. Moreover, on the toolbar, users can clearly know where the current website is hosted (in the Fig. 2, it is hosted in US). This design is considered to be helpful for users, because it is common sense

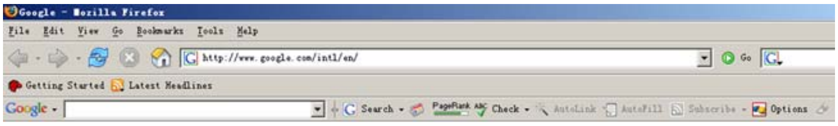


Fig. 1 Google toolbar, Safe Browsing functionality embedded

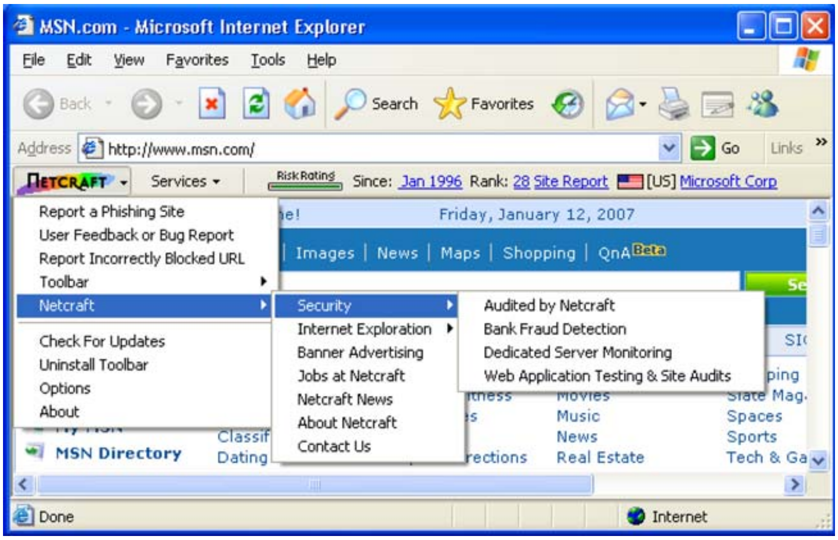


Fig. 2 Netcraft anti-phishing toolbar with a drop-down menu opened

that American Express should not be hosted, for example, in the Middle East countries.

2.3 SpoofGuard

SpoofGuard (Fig 3) is the outcome of one of the researches at Stanford University. It is compatible only with Microsoft Internet Explorer. Compared with the previous two toolbars, this one uses a different type of detection information: the Internet browsing history of the browser. There are three buttons on the toolbar: one for showing the status of the address visited, one for options of the toolbar and one for removing data collected by SpoofGuard (image hashes and password hashes).

SpoofGuard is able to warn about fraudulent web pages by checking the browsing history and other information collected, like domain name, URL, password field, image and links on the page [2]. These five kinds of information are checked in two rounds. In the first round, SpoofGuard finds the similarity between the address to be visited and the browsing history, before the page to be visited is loaded. After the page is loaded, SpoofGuard checks the password input field links on the page and images to assess the similarity of the current page and the pages the user has visited.

After these two rounds, the sum of weighed result values is computed. If the sum is greater than the "Total Alert Level" (a threshold for a warning) the toolbar will warn the user. Moreover, SpoofGuard alerts users, when they try to input the same user identity and password in different web pages. Users can change the criteria from the dialog by pressing the "Options" button. When a warning is shown, two choices are given: continue or stop visiting.

2.4 Anti-phishing IEPlug

Anti-Phishing IEPlug (Fig. 4) is a Microsoft Internet Explorer plug-in completed by the authors of this paper [12]. Likewise SpoofGuard, anti-phishing IEPlug is also a result of a University research project. The idea of this program is that a user maintains a whitelist of those domain names that she/he uses for authenticating critical operations, like e-commerce. Because the whitelist is maintained by the user or computer's system administrator, the developers do not need constant resources for updating of the program. This plug-in is able to alert about forged web pages. After users give the domain names to be detected, the plug-in begins to work. Whenever a page is loaded the plug-in at first checks whether there is a password input field on the page or not. If a password input field is detected, the plug-in will detect whether the address visited contains any domain names saved in the whitelist or not. When the address to be visited includes a keyword that is saved in the whitelist, but the actual domain is different, the plug-in will warn users. For example, a user may save "PayPal.com" to the list of domain names to be protected. When a site containing the keyword is visited, (e.g., http://www.spoofsite.com/paypal/safelogin.htm). The

Fig. 3 SpoofGuard toolbar circled

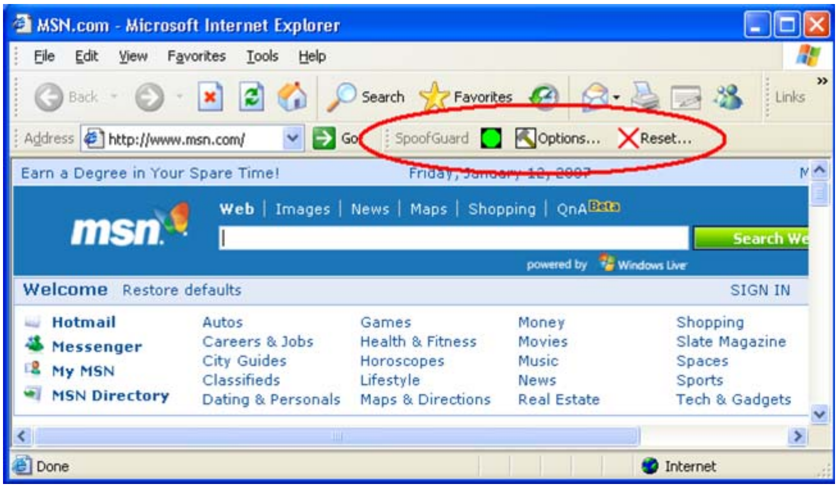
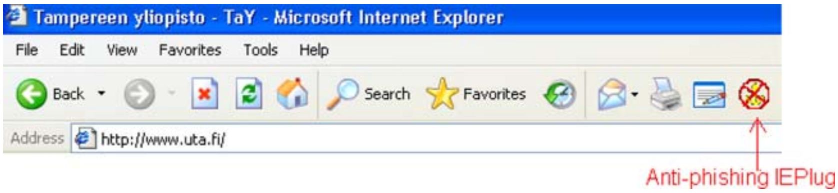

MSN.com - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Links
Address http://www.msn.com/ Go SpoofGuard Options... Reset...
Earn a Degree in Your Spare Time! Friday, January 12, 2007
msn
Web | Images | News | Maps | Shopping | QnA Beta
Search We
powered by Windows Live
Welcome Restore defaults SIGN IN
Hotmail
Messenger
My MSN
MSN Directory
Autos
Careers & Jobs
City Guides
Classifieds
Dating & Personals
Games
Health & Fitness
Horoscopes
Lifestyle
Maps & Directions
Money
Movies
Music
News
Real Estate
Shopping
Slate Magazine
Spaces
Sports
Tech & Gadgets
Done
Internet

Fig. 4 Anti-phishing IEPlug button on the toolbar


Tampereen yliopisto - TaY - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://www.uta.fi/
Anti-phishing IEPlug

IEPlug will detect this link as a possible fraud, because there is a keyword “paypal” and a user is not at the “PayPal.com” website.

In addition, the program actively shows the CA of web pages in the whitelist containing a password field There are two reasons for showing the CA information. On the one hand this shows that a user is at the authentic web page and on the other hand this method educates users.

The web pages can be warned properly based on the domain name list saved on the local machine. The plug-in offers an interface for maintaining these domain names, including adding, editing, and removing. For security reasons limited users can only add domain names to the list.

2.5 Internet Explorer 7 Phishing Filter

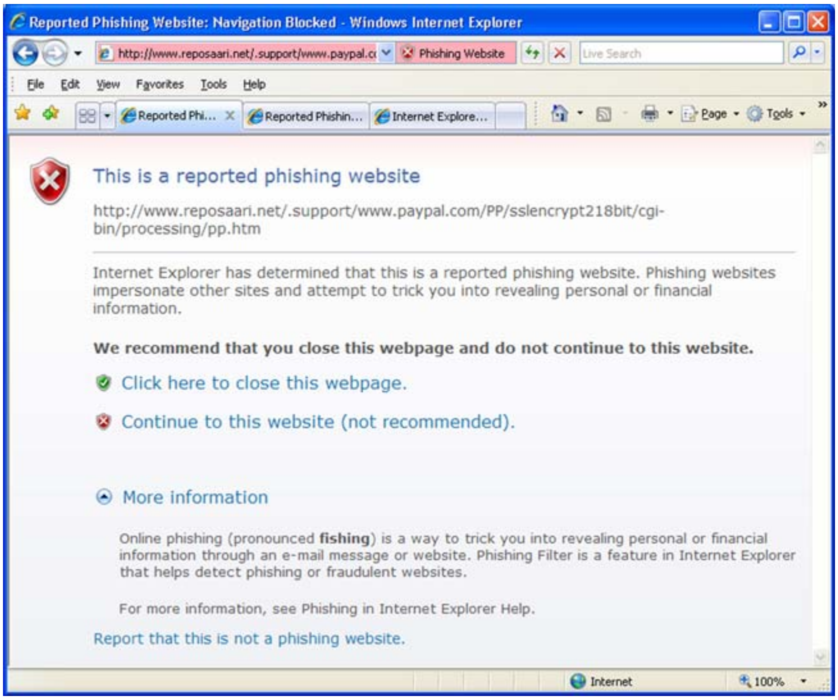
The “Phishing Filter” is a built-in functionality of the IE 7. The filter is based only on the client–server architecture. There are four items in the Phishing Filter’s menu: “Check This Website”, “Turn On/Off Automatic Website Checking”, “Report This Website”, and “Phishing Filter Settings”. The detection mechanism can be described as the following: when a user visits a link (no matter whether it is suspicious or not), IE will firstly send the web address to a Microsoft’s server. Then the server will make a query to the blacklist database and return the detection result back to the client side. If the web page is identified as a spoofing one, the browser will block the attempt to visit a web page (Fig. 5).

3 Heuristic usability evaluation of anti-phishing toolbars

For this usability evaluation, we applied Jakob Nielsen’s heuristic usability evaluation method [14], which is the most common way to inspect software’s usability. There are two reasons why we chose this method. First is that heuristic usability evaluation is flexible and efficient to find out potential usability issues. In addition, this method is helpful and necessary for forthcoming usability tests, because they can be based on the outcome of heuristic evaluation.

Heuristic usability evaluation specifically involves evaluators examining the interface and judging its compliance with recognized usability principles (the “heuristics”) [14]. In other words, the evaluators at first define the heuristics for the user interface to be tested. If the user interface follows the heuristics, that means the interface may be preferable for users. This is a justicial method for each interface evaluated, since the principles are designed based on users’ preferences before the evaluation. Moreover, in order to get reliable results, usability test specialists are required accordingly to the criteria of heuristic usability evaluation.

To guarantee the quality of evaluation results, the minimum number of evaluators is six [14]. Therefore we invited four outside evaluators, who had sufficient working experience either in usability testing or software design. In addition, both of the authors of this paper participated to the evaluation. Because both of us are familiar with phishing prevention, technical context and the design of anti-phishing applications. Furthermore, we know what functionalities



This is a reported phishing website

http://www.reposaari.net/.support/www.paypal.com/PP/sslencrypt218bit/cgi-bin/processing/pp.htm

Internet Explorer has determined that this is a reported phishing website. Phishing websites impersonate other sites and attempt to trick you into revealing personal or financial information.

We recommend that you close this webpage and do not continue to this website.

Click here to close this webpage.

Continue to this website (not recommended).

More information

Online phishing (pronounced **fishing**) is a way to trick you into revealing personal or financial information through an e-mail message or website. Phishing Filter is a feature in Internet Explorer that helps detect phishing or fraudulent websites.

For more information, see Phishing in Internet Explorer Help.

Report that this is not a phishing website.

Fig. 5 A spoofing site is detected by the Internet Explorer’s Phishing Filter

anti-phishing applications should contain. Moreover, we were able to find out potential vulnerabilities when using the toolbars.

The heuristics used are listed in the Appendix A [15]. These heuristics are useful for our evaluation. Firstly, this ensures that each evaluator follows the same principles during the evaluation. In this way, the evaluation can be more reliable. Furthermore, these comprehensive heuristics cover various details about the toolbar interface, which may result in more detailed evaluation results. Following the same items on the list, two evaluation results can be combined together to induce the final testing outcome. In the following parts of this section, we present the methodology, how we designed the heuristic evaluation and what were the results. Afterwards, we summarize the evaluation and what we found during the inspection.

3.1 Detailed design and implementation of heuristic evaluation

Software usability mainly focuses on some specific characteristics of software, including easy to learn (learnability), efficient to use (efficiency), easy to remember, few errors and subjectively pleasing. All of these aspects should be dedicatedly evaluated. Moreover, testing anti-phishing toolbars is not the same as testing other applications. Evaluators have to pay much attention to anti-phishing toolbars' own features during the usability inspection. According to this principle, we designed the heuristic evaluation items carefully.

Evaluation environment. We used one personal computer that was dedicated for testing usability of the anti-phishing applications. In this computer, Firefox 2.0 and Internet Explorer browsers 6.0 were installed. The operating system was Windows XP with all security updates installed. Sometimes, phishing websites contain malicious programs which may compromise the system or interfere the evaluation. Thus, it was necessary to protect the computer. For this evaluation, F-Secure anti-virus client security was installed and the system was backed up to an image file. In case the system would have been compromised it was easy to recover. Furthermore, administrators of the department were aware of the testing, network traffic was monitored, the computer was physically isolated from internal network connections and hardware firewall was present.

There were two monitors installed on the computer, one was the normal screen for the evaluators' and the other was for the observers of the evaluation. The monitors showed the same screen. The phishing websites were collected from the "PhishTank" web site [16]. Because our focus was on usability of toolbars, instead of performance, we picked up the fake sites randomly. This enabled us to see the warnings of each toolbar application in a real environment. More details of the evaluation environment are presented in Tables 1 and 2.

Table 1 Toolbar details

Toolbar	***Version number***	***Download date***
Google Safe Browsing	N/A	13 December 2006
Netcraft toolbar	1.7.0 (20061016)	13 December 2006
SpoofGuard	N/A	Dec 13th 2006
Anti-phishing IEPlug	N/A	10 December 2006
IE 7 Phishing Filter	IE 7.0.5730.11	20 February 2007

Table 2 Evaluation environment

Hardware	
CPU	Pentium III, 800 MHz
Memory	512 MB
Monitors	2 monitors, resolution 1024 × 768, 32 bit color
Keyboard	US-International English keyboard
Software	
Operating system	Windows XP, SP2
Anti-malware product	F-secure Anti-Virus Client Security 5.58
Internet Explorer	IE 6.0, SP2
Firefox	Firefox 2.0

Design of the heuristic evaluation. We collected heuristics following Jakob Nielsen's rules. In terms of these heuristics, a detailed questionnaire (see Appendix A) was implemented.

- *Visibility of the system status.* This heuristic inspects visual capability of the toolbars. Visual capability should be checked in three stages, which are visibility before checking the authenticity of a website, during checking the authenticity of a website, and visibility of the result. In each stage, anti-phishing toolbars should always keep users aware of what is going on and what is the result of identifying the web page. Moreover, response times and types should be reasonable and appropriate.
- *Match between the system and the real world.* Most of vulnerable users do not have enough knowledge of computers and the Internet. From this follows that each operation of the toolbar should be understandable and predictable for non-sophisticated users. This means that people who do not have any professional knowledge about computers and e-commerce should be able to protect themselves based on instructions or warnings from toolbars.
- *User control and freedom.* As mentioned in the previous heuristic rule, we should not expect online commerce customers having learned a lot about computers before. Toolbar designers should not assume that every user can operate each functionality of the toolbar correctly, or as

expected. Furthermore, it is necessary to provide additional functionality to undo and redo what users have done, when they recognize that there is something wrong with their operations. In addition, it should be possible for users to leave the unwanted state before the whole operation completes.

- *Consistency and standards.* This requirement comes from the system requirements. For example, it is difficult to force a Microsoft Windows user to get used to other systems, unless other systems' user interface resembles Microsoft Windows. So is the case with toolbars. The language of the toolbar should follow the platform and browser conventions as well. Moreover, advices should be consistent, when the same risk levels of suspicious web pages or e-mails are detected.
- *Help users recognize, diagnose, and recover from errors.* When users successfully pass the validation to conduct their problematic or incorrect operation, toolbars should also alert or give further advices to correct and recover from errors. This correction should be offered before, during or after users' decisions.
- *Error prevention.* Similar to the third heuristic rule, error prevention can also avoid software failures or potential problems from users' operations. Toolbars are expected to provide necessary check or confirmation before any action is committed. Different from the third heuristic rule, error prevention focuses on the validation of users' each operation and input, instead of undo functionality.
- *Recognition rather than recall.* It is required that any user, no matter who is sophisticated or not, can make a correct decision that prevents phishing without complicated operation sequences. Each warning or advice that a toolbar gives should be understandable enough. In this way users do not need to worry about being compromised due to forgetting correct instructions, even though users are misusing the toolbar.
- *Flexibility and efficiency of use.* In order to prevent phishing, typically users have to make some action, when a fraud attempt is detected. However, sometimes expert users are familiar with how to prevent specific phishing attempts, when a warning comes up and they do not want to read repeated explanations. In this case, it is necessary that flexibility and efficiency of the toolbar can facilitate experienced users' operations, and enable skipping repeated instructions, or conduct the pre-saved default operation. Obviously, flexibility may also result in faulty operation or vulnerabilities. Therefore, it is also required that users are able to return to the default settings.
- *Aesthetic and minimalist design.* This heuristic mainly concentrates on the concision of toolbars' user interface. The task of anti-phishing toolbars is to assist users to identify and stop the fraud, not e.g., commercial promotions. It is meaningful and important to make sure there is only phishing prevention related information in the toolbars. Concise and well-designed toolbar will not confuse users what should be taken into account and what should be done next, when a warning is displayed.
- *Help and documentation.* Users are not omnipotent, and they need to learn how to use different anti-phishing toolbars by themselves. In this case, user manuals, tutorials and instant help should be available with the toolbars.
- *Skills.* Phishers usually take advantage of users' shortage of network or operating system knowledge [4]. Therefore, we expect that toolbars can support, extend, supplement, or enhance users' skills and background knowledge of phishing prevention. Herein, the enhancement should be only resorted from client side, because it is not valuable to evaluate the toolbars' capability against all kinds of phishing techniques.
- *Pleasurable and respectful interaction with the user.* In this heuristic evaluation rule, we try to find out how convenient users experience usage of the phishing prevention toolbar. Both function and aesthetically pleasing value should be considered.
- *Privacy.* Toolbars are used for protecting users' confidential information from being abused or stolen. However, some toolbars also need to know personal information about users, like browsing information and contact details. This kind of information should be also carefully protected.

Besides these heuristic rules, severity of each usability problem should also be defined. Herein, we use the following rating rules provided by Tampere unit for computer–Human interaction (TAUCHI).

1. *Major usability problem*: prevents the users from using the product in a feasible manner and therefore needs to be repaired before the product is launched.
2. *Severe usability problem*: complicates the use significantly and should be repaired immediately.
3. *Minor usability problem*: complicates the use of the product and should be repaired.
4. *Cosmetic usability problem*: should be repaired for the use of the product to be as pleasant as possible.
5. *T. Technical problem*: problems marked with a 'T' are most likely due to technical problems with the product (for example, a feature that has not been implemented yet). Although they are not marked as usability problems, they will be such if left as they currently are.
6. *C. Comment*: comments (or questions) that are used for suggesting operations or point out successful implementations.

Conducting heuristic evaluation. In order to guarantee the quality of evaluation results, the whole procedure of heuristic evaluation was carefully designed. Each evaluator inspected the five toolbars selected. Each evaluator followed the heuristic evaluation criteria presented in the prior section, in addition to the following steps:

1. *Individual preparation.* The selected toolbars were inspected once in average about 2.5 h by each evaluator. The aim was to get a general feeling about each toolbar. Before conducting the evaluation evaluators received the heuristics checklists (Appendix A).
2. *Conducting the evaluation.* The evaluation was conducted at the Virus Research Unit, University of Tampere. We observed each evaluation sequence and our main tasks were to record the findings of each evaluator. At first, Linfeng gave a basic introduction and demonstration of each toolbar in about 15 min. Then, evaluators tested each toolbar based on the preparation and checklist. At this time, they had to explain their findings of usability problems, and their findings were recorded by two authors of this paper. The time for each evaluation is given below. The Phishing Filter was evaluated separately, because reviewers of this paper asked to add the product. Therefore the evaluation time of the Phishing Filter is not included in these times.
   - First evaluator, 2 h (14.12.2006, 13:00–15:00).
   - Second evaluator, 2 h (14.12.2006, 15:00–17:00).
   - Third evaluator, 3 h (15.12.2006, 13:00–16:00). s
   - Fourth evaluator, 3.5 h (15.12.2006, 14:30–18:00).

   We also played the role of evaluators, but the evaluation method was different. When collecting data and writing, we added our ideas and findings to the final results.

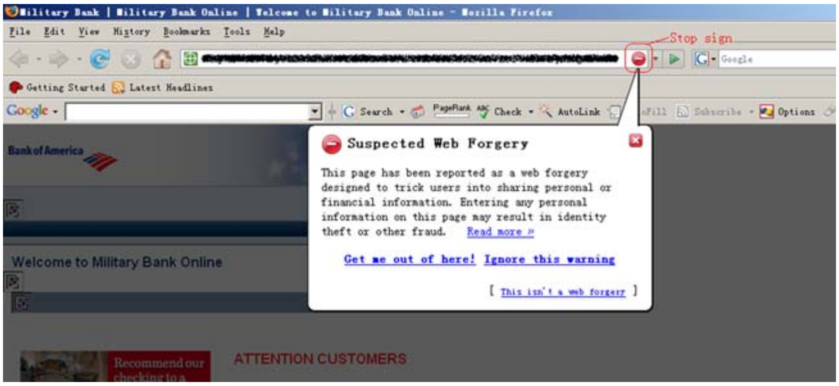
   For the Phishing Filter the evaluation method was the same. The evaluation times were:
   - First evaluator, 0.5 h (20.2.2007, 11.40–12.15).
   - Second evaluator, 0.5 h (26.2.2007, 14:30–15:00).
   - Third evaluator, 0.5 h (26.2.2007, 18:00–18:30).
   - Fourth evaluator, 1 h (26.2.2007, 13:00–14:00).
3. *Gathering evaluation results.* The findings were combined into a single list and the severity of each distinct problem was rated. The following questions were discussed.
   - What were the most severe problem types?
   - What was the overall feeling about the usability of the toolbars?
4. *Reviewing the problem list.* The information from discussion is gathered and summarized for the final evaluation outcome.

4 Discussion

After the evaluation, we gathered the findings from each evaluator. Based on the evaluation and guidance of the checklist (Appendix A), we gained a number of useful usability issues. Please notice, that in heuristic usability evaluation the checklist is meant for guidance, but during the evaluation the evaluators do not need to strictly follow the checklist, but rather to bring forth each usability aspect they will find. We will next discuss the key findings from each toolbar.

4.1 Google Safe Browsing

Good usability design was observed especially when there is a phishing website detected (Fig. 6). "The dimmed area of the browser feels like something, which cannot be accessed. And the balloon can draw user's attention.", an evaluator said. There is no professional terms used, such as phishing or pharming. The advices in the warning, "Get me out of here!" and "Ignore this warning", are understandable. "Any user can easily get the points of them.", said one evaluator. The "Read more" link introduces to web forgery and phishing in technical level. This gives users freedom and is informative. The design principle in of the Google Safe Browsing seems to follow the philosophy of a good security product design. The toolbar shows a clear warning only when a phishing site is detected and otherwise the product remains silent. However, also some usability problems were found during the evaluation. First of all, there are too many functionalities on


Military Bank | Military Bank Online | Welcome to Military Bank Online - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Stop sign
[redacted]
Google
Getting Started Latest Headlines
Google
Search
PageRank
Check
AutoLink
AutoFill
Subscribe
Options
Bank of America
Welcome to Military Bank Online
Recommend our checking to a
ATTENTION CUSTOMERS

Suspected Web Forgery

This page has been reported as a web forgery designed to trick users into sharing personal or financial information. Entering any personal information on this page may result in identity theft or other fraud. Read more »

Get me out of here! Ignore this warning

[This isn't a web forgery]

Fig. 6 A phishing site detected by the Google Safe Browsing

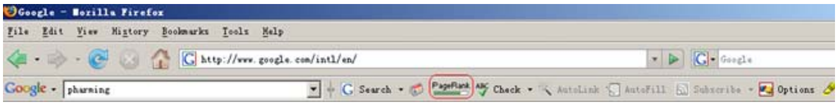
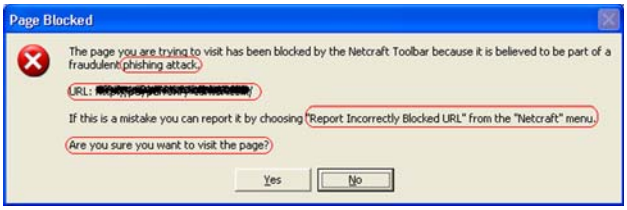


Fig. 7 PageRank functionality encircled



The page you are trying to visit has been blocked by the Netcraft Toolbar because it is believed to be part of a fraudulent phishing attack.

URL: [redacted]/

If this is a mistake you can report it by choosing "Report Incorrectly Blocked URL" from the "Netcraft" menu.

Are you sure you want to visit the page?

Fig. 8 Netcraft anti-phishing toolbar's warning of a phishing site

the toolbar, but no access to the Safe Browsing functionality. It is not obvious that this toolbar can prevent phishing websites. Moreover, some experienced users may misunderstand that the PageRank is part of the Safe Browsing functionality (Fig. 7). In addition, loading the phishing site regardless of the warning (dimmed area in the Fig. 6) may cause some malware stealthily being installed from the visiting phishing website. Moreover, when a user clicks the option, "Ignore the warning", there is no further warning about the danger any more. This is a problem, for example, when a user clicks the choice mistakenly. Furthermore, some evaluators believed that the phishing indicator is not consistent enough. The indicator shows up, only when a phishing website is detected. We also found that when the Google's web site cannot be loaded, (e.g., because of network traffic load) the option "Get me out of here!" will leave the user to the phishing web page, instead of being redirected to the Google's site. As a consequence a user may erroneously believe to be in a safe site. Finally, one evaluator was concerned that the warning icon (circled in Fig. 6) may be confused with the lock icon at the address bar of the Firefox browser.

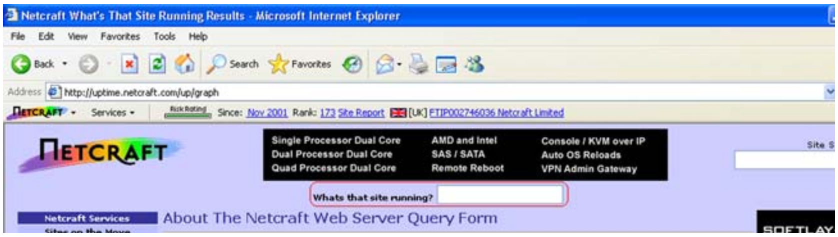
4.2 Netcraft anti-phishing toolbar

Compared to the Google toolbar, Netcraft anti-phishing toolbar is designed only for phishing prevention. Therefore the toolbar's user interface is straightforward. The information about the website visited is displayed on the toolbar directly. The online tutorials are well designed.

However, the information present on the toolbar is not easily understandable. The most obvious usability problem is the implementation of the two drop-down menus "Netcraft" and "Services". Some useful options are placed unexpectedly and inconsistently. For example, "Report a phishing site" and "Report Incorrectly Blocked URL" should be services, but they are found from the "Netcraft" menu. Furthermore, structure of the menus is too complex to be easily understandable.

There are also some other minor usability issues, which may cause ambiguity. For instance, the toolbar item, "Since", is right after the "RiskRating", which may confuse users; the criteria of "Rank" is imprecise; "Site Report" does not tell whether a site is fraudulent or not. Rather the site report shows technical server information. Not every user understands the information or needs it, especially normal end-users. When a phishing website is detected, inexperienced users may not understand information in the warning dialog (Fig. 8). First of all, the popup dialog is similar to a website dialog or operating system dialog (e.g., illegal memory reference). Secondly, toolbar designers should not assume every user knows these professional terms, such as phishing and URL. Furthermore, it is not necessary to show the URL, because not every user understands the expression, especially if the web address is complex. In addition, "Report Incorrectly Blocked URL", highlighted in the warning dialog, should be clickable in order to encourage users for submitting reports, instead of forcing users to remember what they should do and where they can find the functionality. Finally, the expression "Are you sure you want to visit the page" is not clear enough. When a user quickly reads this she or he may click a wrong button. Similarly, when there is a suspicious website detected, there is no advice, except the color change of the "RiskRating" indicator.

Netcraft toolbar has a powerful website to support its services. Some of the important functionalities, such as reporting, have to resort to the website. Therefore the related web pages should be evaluated as well. As the program relies on web pages, problems will appear when there is network load or the web pages are not available. This should be taken care of in the product. For example, there could be internal help system, in addition to the web page help system. One typical usability problem is that the input field of "What's that site running" is not distinguished enough (Fig. 9). First of all, the input field is buried under other more distinguished text and advertisements. The input field can hardly draw users' attention. In addition, there is no "Submit" button. The toolbar configuration settings (Fig. 10) are not designed well from the usability perspective. First of all, the options are not grouped appropriately. For example, the appearance, and the functional settings (e.g., the level of automatic blocking) should be separated; the "Remember Details for Report URL Form" option should be more clearly grouped together with the Name and E-mail fields.


Netcraft What's That Site Running Results - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
NETCRAFT
Services
Risk Rating
Site S

http://uptime.netcraft.com/up/graph

Since: Nov 2001 Rank: 173 Site Report [UK] FTIP002746036 Netcraft Limited

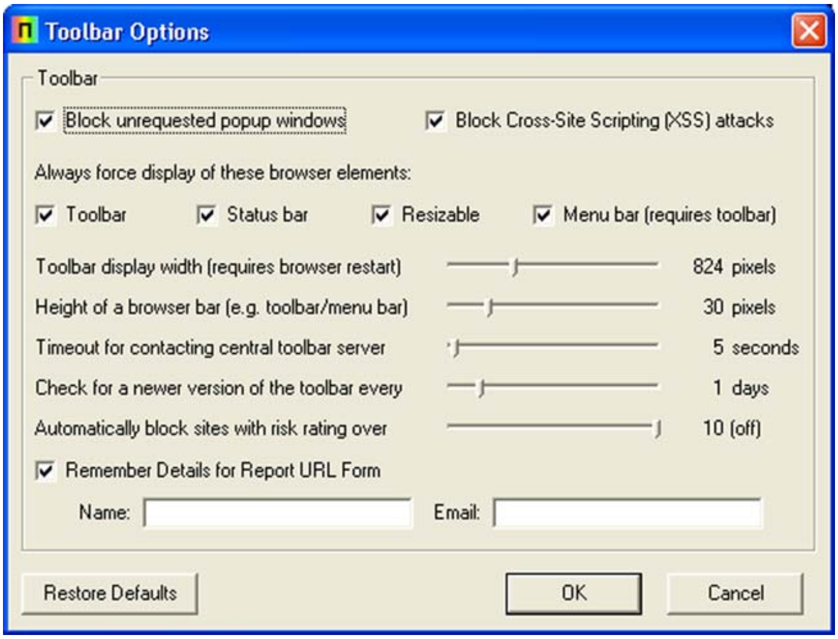
Single Processor Dual Core AMD and Intel Console / KVM over IP
Dual Processor Dual Core SAS / SATA Auto OS Reloads
Quad Processor Dual Core Remote Reboot VPN Admin Gateway

Whats that site running?

Netcraft Services

About The Netcraft Web Server Query Form

Fig. 9 Not well designed input field at the Netcraft website



Toolbar

☑ Block unrequested popup windows ☑ Block Cross-Site Scripting (XSS) attacks

Always force display of these browser elements:

☑ Toolbar ☑ Status bar ☑ Resizable ☑ Menu bar (requires toolbar)

Toolbar display width (requires browser restart) 824 pixels

Height of a browser bar (e.g. toolbar/menu bar) 30 pixels

Timeout for contacting central toolbar server 5 seconds

Check for a newer version of the toolbar every 1 days

Automatically block sites with risk rating over 10 (off)

☑ Remember Details for Report URL Form

Name: Email:

Fig. 10 Options of the Netcraft toolbar

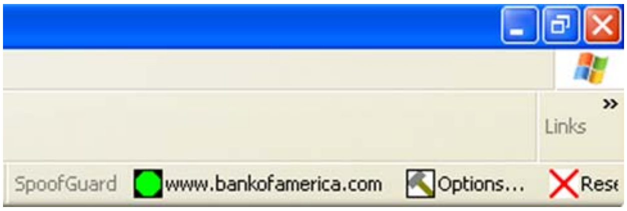
In addition, the controls used do not follow common sense, such as checking for a newer program version.

4.3 SpoofGuard

Some advantages in the user interface were found. The traffic light is consistent enough in order to help users for identifying the risk of the current web page. Furthermore, the toolbar keeps user informed and the design is clear and concise enough.

However, from the usability point of view, there are some places to be improved. Firstly, if the web address is long enough, the Reset button and Options may be pushed outside the screen (Fig. 11).

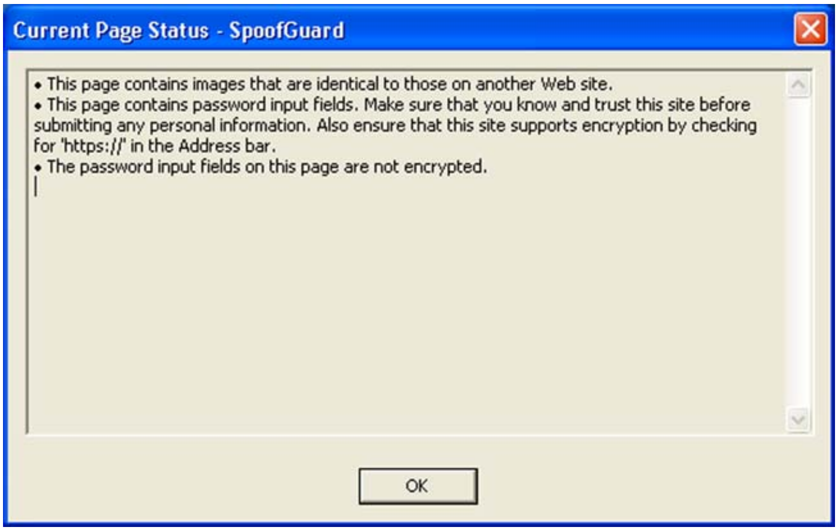
The indicator showing identity of a web page is the color of the traffic light. However, this may be an obstacle for color-blind users. In addition, there are cultural differences as, for example, in India the color codes are different. Furthermore, the function of the reset button is unclear. Nothing seems to happen, when a user clicks the button. Users may also confuse the “Reset” button with resetting the options to the default values, but actually clicking the button will remove the image hashes and password hashes. Moreover, while the red cross refers to deleting something, it is unobvious that clicking the button resets the configuration data. Similarly, the suggestion for users is not explicit enough, when a suspicious web page is detected (Fig. 12). The suggestion is likely to confuse inexperienced users, when they want to know whether they should trust the web page or not.



www.bankofamerica.com

Fig. 11 Too long domain name in SpoofGuard

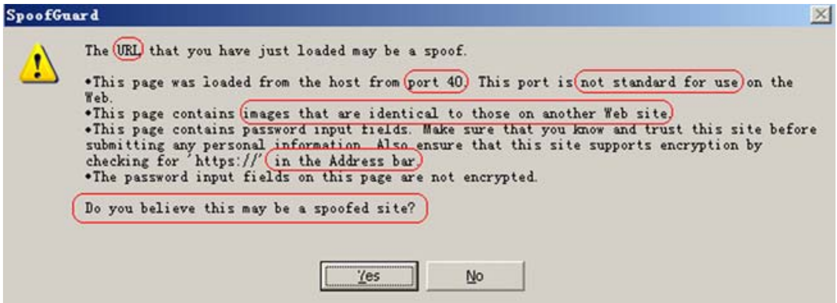
At last, there are some comments about the warning when a spoof web page is detected (Fig. 13). Similar to those comments about the Netcraft toolbar, some terms are too professional to be understood by normal users. Furthermore, when SpoofGuard lists suspicious places of a web page, users should be able to learn more about them, e.g., why cannot images be identical to those on another web site? Finally,



Current Page Status - SpoofGuard

- This page contains images that are identical to those on another Web site.
- This page contains password input fields. Make sure that you know and trust this site before submitting any personal information. Also ensure that this site supports encryption by checking for 'https://' in the Address bar.
- The password input fields on this page are not encrypted.

Fig. 12 Information about a suspicious web page



SpoofGuard

The URL that you have just loaded may be a spoof.

•This page was loaded from the host from port 40. This port is not standard for use on the Web.
•This page contains images that are identical to those on another Web site.
•This page contains password input fields. Make sure that you know and trust this site before submitting any personal information. Also ensure that this site supports encryption by checking for 'https://' in the Address bar.
•The password input fields on this page are not encrypted.

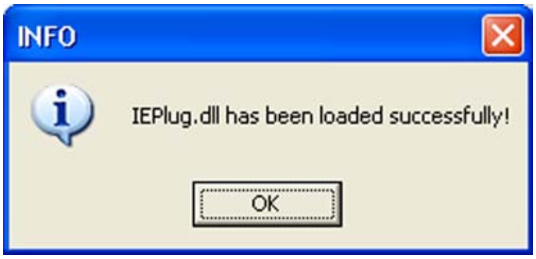
Do you believe this may be a spoofed site?

Fig. 13 Warning of a spoof web site

users can hardly understand what will happen when they click the "Yes" or "No" button. The question should be clearer.

4.4 Anti-phishing IEPlug

Likewise SpoofGuard, Anti-phishing IEPlug is also a result of a University research project. Compared to the previous three toolbars, the user interface of this application on Internet Explorer toolbar is very simple, only one button. Furthermore, the idea to maintain only a whitelist was considered convenient, because it gives power to users and does not require constant updating. However, some parts of the program needed to be improved. The popup messages were used too frequently. Sometimes, they were annoying. For example, when Anti-phishing IEPlug is installed successfully, a message was shown each time Internet Explorer is opened (Fig. 14). The reason for showing the dialog was to show a user that the program is active and protecting the user. A better solution would be to show the program status e.g., as an icon on the toolbar. Another example is that when Anti-phishing IEPlug adds a domain name, there is no need to remind users with a popup message. Furthermore, the message text is too technical. End-users are likely to be confused.

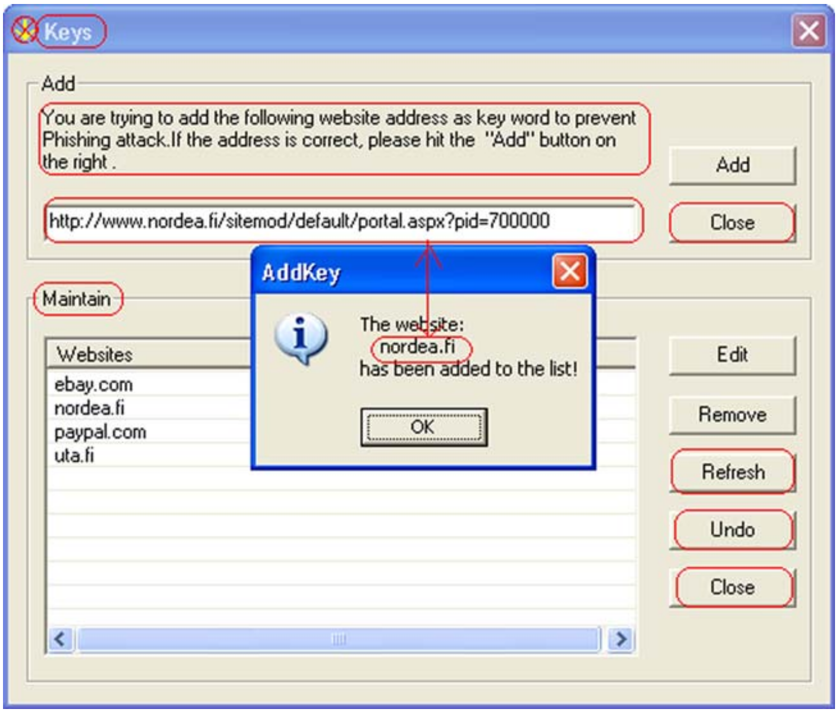


INFO

IEPlug.dll has been loaded successfully!

Fig. 14 Popup message of the Anti-phishing IEPlug

The interface of the domain name configuration (whitelist) dialog was not satisfactory either (Fig. 15). First of all, the title of the dialog does not make sense. A title should represent the purpose of the dialog. Likewise with other toolbars, some terms were too difficult to understand. The edit dialog for domain names was not long enough and adding domain names was not consistent. The address was shown in the edit box, but only domain name was added to the whitelist. Furthermore, the dialog should be stretchable so that users can view as many domain names as possible at one glance. The buttons were not well designed. There were two "Close"

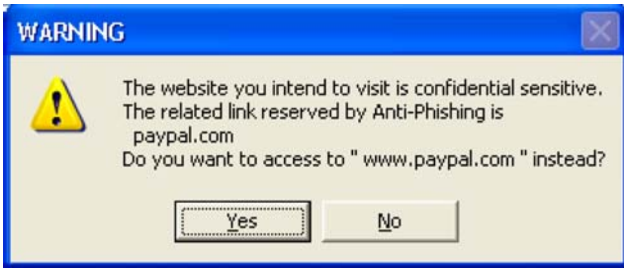


You are trying to add the following website address as key word to prevent Phishing attack.If the address is correct, please hit the "Add" button on the right .

http://www.nordea.fi/sitemod/default/portal.aspx?pid=700000

The website:
nordea.fi
has been added to the list!

Fig. 15 Adding a domain name to the whitelist



The website you intend to visit is confidential sensitive.
The related link reserved by Anti-Phishing is
paypal.com
Do you want to access to " www.paypal.com " instead?

Fig. 16 Warning of the Anti-phishing IEPlug

buttons and there was no feedback, when a user clicks "Refresh" or "Undo".

The warnings of the Anti-phishing IEPlug were also problematic from the usability point of view (Fig. 16). There were too many technical terms and there was no further information available to users. There should be a well designed help system to give users more information. In addition, it would be convenient that popular authentic domain names were pre-saved to the whitelist. This feature could be implemented with client–server architecture, by collecting websites from user's whitelists.

4.5 Internet Explorer 7 Phishing Filter

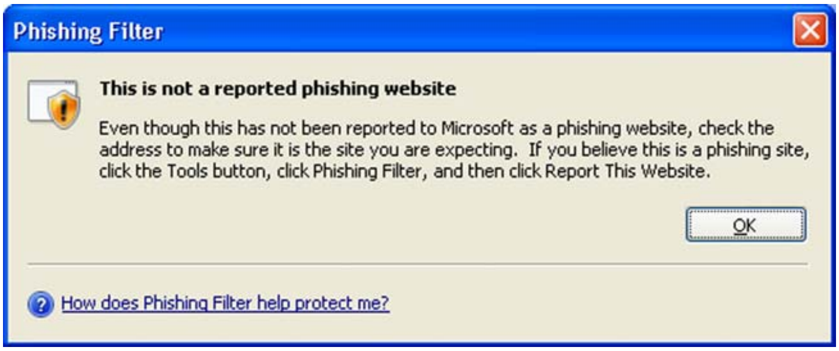
The Phishing Filter is embedded functionality of the IE 7. This kind of design should be able to co-operate with other components of the IE 7. The pros of Phishing Filter include colored address bar (a constant warning indicator), straight and informative warning, off-line help documentation, as well as a well designed interface to report suspicious and falsely detected web sites. However, there are also some places to be improved. In the following, we will present the usability problems found during evaluation.

The warning can successfully stop users from visiting identified phishing web pages (Fig. 5), but not everything is satisfactory. At first, the icons for two options on warning page are not evident enough. These icons are the same as those in the dialog "Turn On/Off Automatic Website Checking". However, the functionalities are not related. Moreover, it would be better to give the criteria for phishing site detection.

A serious usability issue is that when a user clicks the choice "Click here to close this webpage" (Fig. 5), there is a confirmation that prevent users from closing the web page directly. This design may mislead users to think their action is risky. However, there is no extra warning when a user clicks the choice to visit the phishing web site detected.

"Check The Website" is a sub-functionality of the Phishing Filter, which can report the authenticity of the current web page. However, the information given is not good enough. First of all, when the current page is not in the black list, it cannot tell users how to check the authenticity manually. Furthermore, the instruction of how to "Report This Website" is not easy to remember. Instead, a link should be given to make users report the current page.

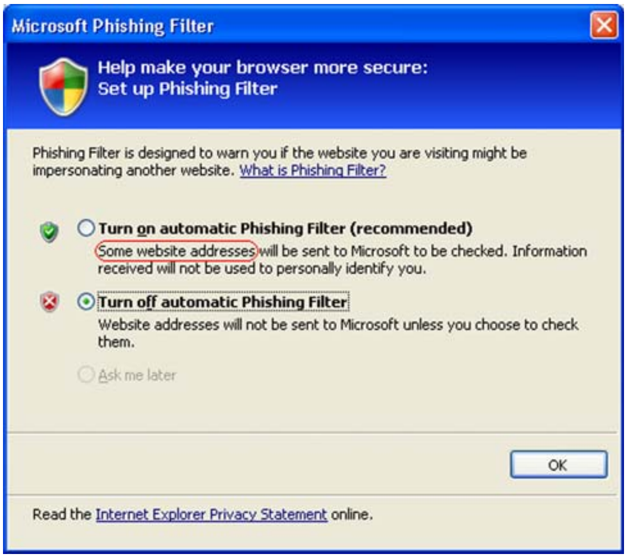
One serious security related problem we found is that the dialog (Fig. 17) is still shown, even when the network connection is disabled. This will mislead users and what is worse,


Phishing Filter

This is not a reported phishing website

Even though this has not been reported to Microsoft as a phishing website, check the address to make sure it is the site you are expecting. If you believe this is a phishing site, click the Tools button, click Phishing Filter, and then click Report This Website.

Fig. 17 The dialog when a user clicks "Check The Website" and it is not a reported phishing website


Microsoft Phishing Filter

Help make your browser more secure:
Set up Phishing Filter

Phishing Filter is designed to warn you if the website you are visiting might be impersonating another website. What is Phishing Filter?

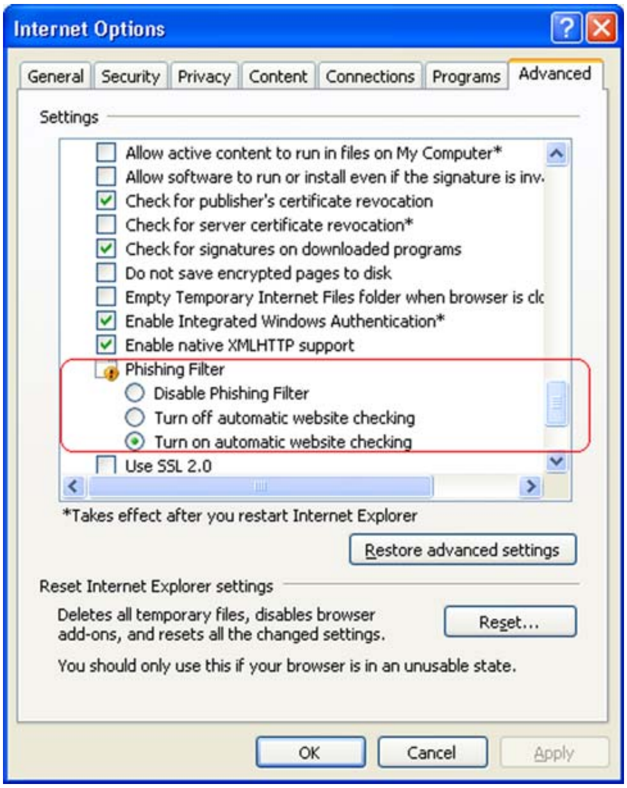
○ **Turn on automatic Phishing Filter (recommended)**
Some website addresses will be sent to Microsoft to be checked. Information received will not be used to personally identify you.

● **Turn off automatic Phishing Filter**
Website addresses will not be sent to Microsoft unless you choose to check them.

○ Ask me later

Read the Internet Explorer Privacy Statement online.

Fig. 18 The dialog to "Turn off the Automatic Website Checking" while the current setting is ON


Internet Options

Settings

- ☐ Allow active content to run in files on My Computer*
- ☐ Allow software to run or install even if the signature is inv
- ☑ Check for publisher's certificate revocation
- ☐ Check for server certificate revocation*
- ☑ Check for signatures on downloaded programs
- ☐ Do not save encrypted pages to disk
- ☐ Empty Temporary Internet Files folder when browser is clo
- ☑ Enable Integrated Windows Authentication*
- ☑ Enable native XMLHTTP support
- Phishing Filter
  - ○ Disable Phishing Filter
  - ○ Turn off automatic website checking
  - ● Turn on automatic website checking
- ☐ Use SSL 2.0

*Takes effect after you restart Internet Explorer

Reset Internet Explorer settings

Deletes all temporary files, disables browser add-ons, and resets all the changed settings.

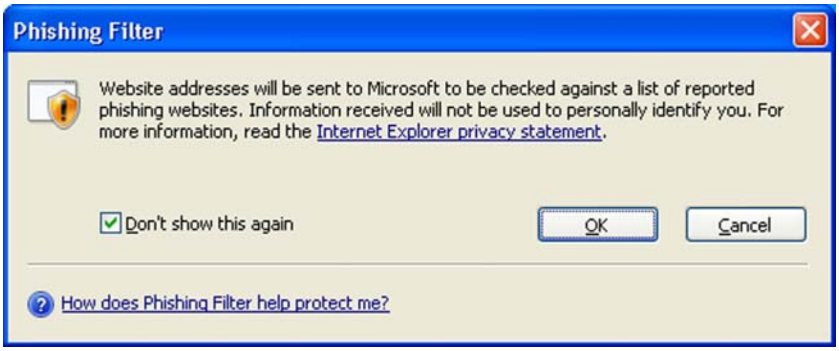
You should only use this if your browser is in an unusable state.

Fig. 19 The dialog to change the settings of the Phishing Filter. The settings of the Phishing Filter are circled

if the network connection fails, it seems that the Phishing Filter will give false advice to a user. "Turn On/Off Automatic Website Checking" is problematic from the usability point of view. First of all users are likely to confuse this functionality with the "Automatic Website Checking" setting. Secondly, when a user wants to turn off the automatic checking (Fig. 18), the status seems to be already at "Turn off…". This design in meant to facilitate the operation, but users are likely to be confused.

Moreover, the instruction on this dialog is also ambiguous. For example, it is said "Some website addresses…" (circled in Fig. 18). A user may wonder why not all addresses will be sent. This may make users feel insecure. A better solution would be to show these choices with the "Phishing Filter options" along with all the other possible choices. Furthermore, there are more than these three settings for the Phishing Filter. All settings should be present in one dialog. The menu item "Report This Website" allows a user to submit a suspected phishing web site to the server at Microsoft. There are three steps, in order to successfully submit. Even though users may understand these steps, there are some minor problems. For example, there is no way to recover a mistaken submission. Moreover, there is too little information about how this submission works or helps other users. There is a challenge shown to the user, but the letters were sometimes difficult to interpret correctly.

"Phishing Filter Settings" allows a user to switch on/off automatic checking and to disable the filter. However, when a user clicks this functionality, a general "Internet Options" dialog is shown (Fig. 19), instead of the Phishing Filter's setting dialog. Furthermore, there is a long list on this dialog,



Phishing Filter

Website addresses will be sent to Microsoft to be checked against a list of reported phishing websites. Information received will not be used to personally identify you. For more information, read the Internet Explorer privacy statement.

☑ Don't show this again

Fig. 20 The dialog shown when a user clicks "Check This Website"

and the settings for the Phishing Filter are listed nearly at the end of this list. This design really disturbs users. It would be better to put these settings to a separate options dialog. The privacy issue is also taken good consideration by Microsoft. When a user uses "Check The Website" of Phishing Filter for the first time, the "Internet Explorer privacy statement" is shown explicitly (Fig. 20). However, it would be possible to store the black list locally in a similar way as the "Google Safe Browsing" functionality allows. In this way users could have more trust on privacy protection, because there would be no need to send browsing information to an external server. There are also some general usability problems. The Phishing Filter may not be easy to find from the "Tools" menu. Furthermore, there is no direct entrance or button to the "Check This Website" functionality. It is not even possible to add a button to a toolbar. In addition, the help documentation should be accessible from the sub-menu of the "Phishing Filter".

4.6 Statistics

After the evaluation, we firstly reviewed the comments and then completed the final heuristic checklist. Based on the heuristic usability criteria, we assigned the severity level to each usability problem. After that, we collected the usability problems of each anti-phishing application and constructed the following statistics tables. According to the Nielsen's principles, six evaluators are sufficient to find most usability problems [14]. Therefore, the sample size is large enough for the heuristic usability evaluation.

We cannot give detailed heuristic evaluation results, because of the length limitation. Therefore we present general statistics of the usability problems found and their severity levels. The statistics are not meant for comparing the toolbars' usability performance. Instead we want to show that there exists a number of usability problems in the toolbars evaluated.

We would like to advice the reader that there are also limitations with these statistics. First of all, these statistics are very rough evaluation results, which cannot reflect every usability problem precisely. Even though the heuristic checklist was designed beforehand, the entire evaluation is based on the evaluators' personal opinion. Therefore the result of the evaluation may not be comprehensive enough. For further research, it is necessary to conduct a usability testing in order to collect users' experiences and feedbacks directly. The statistics for their evaluation results are listed below (Tables 3, 4, 5, 6, 7):

Table 3 Statistics for Google Safe Browsing

	Major	Severe	Minor	Cosmetic
Visibility	2	1	0	3
Matching the real world	1	1	1	0
User control & freedom	1	3	1	0
Consistency & standards	0	1	1	0
Help user recognize	1	0	0	1
Error prevention	0	1	0	0
Recognition	2	0	0	0
Flexibility	1	3	1	0
Aesthetic design	0	0	0	0
Help & documentation	1	1	2	1
Skills	1	2	1	0
Pleasurable interaction	0	1	0	0
Privacy	0	0	0	0

4.7 Suggestions for improving usability of toolbars

According to the comments and statistics constructed from the evaluation, we made a number of findings for anti-phishing client side application usability design. Generally, we found that there are three basic components that should be well designed: the main user interface of the toolbar, warnings, and help system. We will next discuss our key findings in these components.

Table 4 Statistics for Netcraft anti-phishing toolbar

	Major	Severe	Minor	Cosmetic
Visibility	3	5	0	0
Matching the real world	5	5	0	0
User control & freedom	1	2	1	0
Consistency & standards	3	4	0	1
Help user recognize	3	0	0	0
Error prevention	2	1	0	0
Recognition	3	0	0	0
Flexibility	0	5	1	0
Aesthetic design	1	2	0	0
Help & documentation	0	1	1	0
Skills	2	1	1	0
Pleasurable interaction	0	3	0	0
Privacy	0	1	0	0

Table 5 Statistics for SpoofGuard

	Major	Severe	Minor	Cosmetic
Visibility	1	1	0	1
Matching the real world	4	2	0	0
User control & freedom	1	2	0	0
Consistency & standards	2	3	0	0
Help user recognize	1	1	0	0
Error prevention	2	0	0	0
Recognition	0	2	0	0
Flexibility	0	5	1	0
Aesthetic design	1	0	0	0
Help & documentation	5	5	0	0
Skills	2	1	2	0
Pleasurable interaction	1	2	0	0
Privacy	2	0	0	0

Table 6 Statistics for Anti-phishing IEPlug

	Major	Severe	Minor	Cosmetic
Visibility	1	5	3	0
Matching the real world	1	4	0	0
User control & freedom	1	2	0	0
Consistency & standards	3	2	0	0
Help user recognize	1	0	0	0
Error prevention	1	1	0	0
Recognition	2	3	0	0
Flexibility	0	4	2	0
Aesthetic design	1	2	0	0
Help & documentation	3	6	1	0
Skills	3	2	0	0
Pleasurable interaction	1	3	0	0
Privacy	0	0	0	0

Table 7 Statistics for IE7 Phishing Filter

	Major	Severe	Minor	Cosmetic
Visibility	3	2	1	0
Matching the real world	1	2	0	0
User Control & freedom	2	2	1	0
Consistency & standards	1	0	0	1
Help user recognize	2	0	0	0
Error prevention	1	0	0	0
Recognition	3	1	0	0
Flexibility	3	2	0	1
Aesthetic design	1	0	1	0
Help & documentation	1	2	1	0
Skills	2	3	0	0
Pleasurable interaction	1	2	0	0
Privacy	0	0	0	0

1. *Main user interface of the toolbar.* According to our perceptions, the main user interface of the toolbar is very important. First of all, the status of the toolbar should be shown appropriately. This means that whenever browsing a web page, the user should be able to easily observe what toolbar is doing and whether the current web page is authentic or not. Secondly, the anti-phishing client side application interface should be simple enough so that it is easy to understand and it does not take too much space from the browser's interface. Of course, frequently used and important functionalities, such as configuration settings and viewing the website identity analysis result, reporting a suspicious or misjudged web page, should be convenient enough to be found. In this regard, some parts of the SpoofGuard's user interface design could be a very good example, such as the traffic light indicator and the Options button. These buttons are informative and make functionalities easily accessible.
2. *Warnings.* Considering the lack of reliable strategy to detect the fraud, the warnings of the application need to be carefully designed. It is important that a user is able to react correctly when a fraud or suspicious web page is found. According to the evaluation by Zhang et al. [19] observation, the false and undetermined detection is not a minor issue. It would be problematic if a user relies only on these toolbars with fixed detection algorithms. Therefore, there should be at least three levels of security indication: the warning for detected web forgery, the warning for a determined suspicious page and the indication for innocent or authentic page.

   The warning of the Google Safe Browsing is a good example for showing the web forgery. The Google's warning can stop users' faulty visits properly. The warning for suspicious page can be the same as the one for the forgery. The differences between them could be on the given advices and their indications. For example, there may be only one advice (stop visiting) available for the forgery, and the indicator for the warning could be a stop sign. Furthermore, there could be two further advices (stop visiting, or check authenticity manually), when a suspicious page is found. The indicator should not be as strong as the one for detected page,(e.g., an exclamatory mark). The undetermined page requires to be notified to users as well. When this kind of page is found, the instant help documentation or instructions are needed in order to help users identify suspicious pages manually. Additionally, in order to be consistent innocent pages should be indicated, respectively. For instance, the indicator could be shown at the same location of other levels of phishing warning indicators. Finally, a double warning should be used in case an erroneous choice is made. If a user accidentally selects a choice that leads to visiting a phishing website, the second warning should be available to correct the mistake.

3. *Help system.* Compared to other software, the client side anti-phishing application must be able to help users at any critical occasion. These occasions include when users may select a dangerous choice, when they are confused by some terms, and when they want to learn how to identify a correct service manually. Regarding the efficiency and convenience of help, the ways of showing help for different occasions may not be the same. For example, when a user tries to find further advice, instant help system is needed. Another example is when a user should find out consequences of different choices when a warning is present. However, the text, which may help the user to understand some terms or consequences of choices, cannot be put together with the warning. It must be remembered that too much information will confuse users. For the other two occasions, the online help documentation would be better, because there can be much more information. The help system of the Netcraft toolbar could be a valuable example.

Finally we have two general results. First of all, it is beneficial to apply whitelist and blacklist methods together. Even though blacklist based application is able to correctly detect the identified (or reported) phishing sites, it is still possible to fail to warn non-identified or non-reported ones. Relatively, whitelist contains information of the websites to be protected, but all phishing websites cannot be identified. Therefore, more protection could be gained by combining these two kinds of lists into an anti-phishing application. Actually this is currently possible by using a blacklist based application together with a whitelist based application. For example, the Google Sage Browsing, Netcraft or Phishing Filter could be used together with the Anti-phishing IEPlug.

The other general finding is that, anti-phishing client side applications should not rely merely on the Internet, because sometimes the online traffic is not good enough. For example, when a user chooses an option to leave a phishing website, a toolbar could direct the user to a safe page. However, if the connection fails at that time (we met this occasion during the evaluation), the user may stay on the fraud website. This places the user at unnecessary risk. Therefore, anti-phishing applications had better redirect to the locally saved page or in some other way handle the possible fault with the Internet connection. Furthermore, it should be taken care of that the online help systems and reporting systems, which rely on the Internet connection, may not work all the time.

5 Conclusions

In this paper, we presented a design of heuristic evaluation of five typical anti-phishing applications, and discussed our findings from the evaluation. As far as we know, this evaluation is novel usability research in the phishing prevention domain. We found some important usability issues, which could be helpful for further improvement of anti-phishing toolbars. Furthermore, the heuristics checklist could be reusable for future testing as well.

However, there are also some limitations in the evaluation. Due to the natural drawbacks of heuristic evaluation, we cannot get precise and direct users' feelings on using these toolbars. Moreover, because of the limited number of evaluators, not every usability problem was found. With conducting the future usability test, those drawbacks could be overcome with larger resources. In addition, we failed to indicate whether showing actively in our application the CA is usable or not. Finally, the number of evaluated application types is limited. When we were conducting the evaluation, we realized that there could be more than four types of anti-phishing client side applications.

Despite of the limitations of this evaluation, there are some contributions to the anti-phishing research domain. We succeeded to construct a heuristic checklist, and found out some key usability issues based on the evaluation, including how to implement an appropriate warning, how to warn a user in an understandable and polite manner, and what are essential components of client-side anti-phishing applications. All of these may facilitate the future usability design and be a basic guidance in the anti-phishing usability domain.

Acknowledgements We would like to thank the evaluators Lauri Immonen, Lily Wen Lin-Marsalo, Wenfeng Liu and Markku Myllylahti for their participation to the evaluation and for their effort and valuable input for our paper. Without their help our paper would not have been possible. We would also like to thank the EICAR program committee and especially Vlasti Broucek for valuable and constructive comments to our paper.

A Appendix A: Toolbars heuristic evaluation: checklist

A.1 Visibility of toolbar status

The toolbar should always keep users informed about what is going on, through appropriate feedback within reasonable time (Table 8).

A.2 Match between toolbar and the real world

The toolbar should speak the user's language, with words, phrases and concepts familiar to the user, rather than software oriented terms. Follow real-world conventions, making information appear in a natural and logical order (Table 9).

A.3 User control and freedom

Users should be free to select and sequence tasks (when appropriate), rather than having the toolbar do this for them. Users often choose toolbar functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Users should make their own decisions (with clear information) regarding the costs of exiting current work. The toolbar should support undo and redo, when user choose advices given by toolbars (Table 10).

A.4 Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions (Table 11).

A.5 Help users recognize, diagnose, and recover from errors

Error messages should be expressed in plain language (NO CODES; Table 12).

A.6 Error prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place Table 13).

Table 8 Visibility of toolbar status checklist

#	Review checklist	Yes No N/A	Comments
1.1	Does every display begin with a title or header that identify itself?		
1.2	Is the toolbar status shown before visiting any new web page?		
1.3	Is it easy to find which operations are available before visiting any web page?		
1.4	Is the toolbar status shown during verifying legitimacy of web pages?		
1.5	Is there any suggestion on what user should do during waiting for verification result?		
1.6	If there are observable delays (greater than fifteen seconds) in the toolbar's verification response time, is the user kept informed of the toolbar's progress?		
1.7	Is the web page legitimacy analysis result shown properly after showing the content of web page?		
1.8	If error occurs because of users' mis-operation, is user able to see the field in error?		
1.9	Is there some distinguished form of toolbar feedback for warning, when the fraud is detected?		
1.10	Is it clear to know which items in the dialog or toolbar are selectable?		
1.11	Are every two items separated properly on the toolbar?		
1.12	Are the buttons on the toolbar separated obviously?		
1.13	Is there any clear explanation from toolbar before significant operation (e.g., decide to keep visiting the warned suspicious web page)?		
1.14	Are the fraud detection response time less than 1 s, regardless connection delay?		
1.15	Is the used terminology consistent with anti-phishing domain?		
1.16	Can image on button express the function of it correctly?		
1.17	Can user distinguish different GUI controls from each other (e.g., Drop-down list does not look like a button) ?		
1.18	Can users know what operations are available, when the fraud is found?		
1.19	Can users know whether the visiting web page is deceptive one or not, after toolbar's verification?		

Table 9 Match between toolbar and the real world checklist

#	Review checklist	Yes No N/A	Comments
2.1	Are icons meaningful and concrete?		
2.2	Are the menu items and buttons on the toolbar ordered in the logic way, giving users what will be done after selection?		
2.3	If there is a visual cue (e.g., images on buttons), does it follow real-world conventions?		
2.4	Are there any obvious differences between selected and unselected?		
2.5	On user input field, are tasks described in terminology familiar to users?		
2.6	Are the questions understandable, which are given by popups of toolbar?		
2.7	Do menu choices or words on buttons have readily understood meanings?		
2.8	Are menu items parallel grammatically?		
2.9	Do commands follow the language in daily life, instead of computer science domain?		
2.10	If in need of input, is it available to give uncommon letters?		
2.11	Is key function of toolbar labeled clearly and distinctively?		
2.12	Are fields on the window separated appropriately?		
2.13	Are fields on the window grouped appropriately?		
2.14	Are buttons or menu items grouped appropriately?		

Table 10 User control and freedom checklist

#	Review checklist	Yes No N/A	Comments
3.1	Can users check legitimacy of web page at any time when they want?		
3.2	Can users conduct several same functional operations together, instead of repeating them one by one?		
3.3	Is there "Undo" function provided, or can user cancel the former operation which can cause serious consequence?		
3.4	Are menus on the toolbar broad (many items on a menu) rather than deep (many menu levels)?		
3.5	When manipulating sensitive data (e.g., delete or add the fraud web page name from black list), can any user have the access?		
3.6	Can users set their own preferred layout of toolbar?		
3.7	Can users set their own preferred warning methods?		

A.7 Recognition rather than recall

Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of toolbar should be visible or easily retrievable whenever appropriate (Table 14).

A.8 Flexibility and Minimalist Design

Accelerators-unseen by the novice user-may often speed up the interaction for the expert user such that the toolbar can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. Provide alternative means of access and operation for users who differ from the "average" user (e.g., physical or cognitive ability, culture, language, etc.; Table 15).

A.9 Aesthetic and minimalist design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility (Table 16).

A.10 Help and documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search,

Table 11 Consistency and standards checklist

#	Review checklist	Yes No N/A	Comments
4.1	Are the locations of buttons on toolbar in the browser fixed and consistent?		
4.2	Is the terminology used consistently?		
4.3	Are icons labeled clearly?		
4.4	Are the titles of popups consistent?		
4.5	Is color tone used on toolbar consistent with browser?		
4.6	Does the menu structure and buttons layout match the task structure?		
4.7	Do online instructions appear in the consistent place?		
4.8	Do toolbar error messages follow hosted browsers' message standards?		
4.9	Do toolbar warnings follow hosted browsers' warning standards?		
4.10	Are attention-getting techniques used with care?		
4.11	Is the attention-getting technique used only for warning of the fraud detection or also for exceptional conditions?		
4.12	Is the most important information placed at the beginning of the prompt?		
4.13	Are advices for users named consistently across all prompts in the toolbar, no matter what the risk level of warning is?		
4.14	Does the structure of menu items' names match their corresponding contents?		
4.15	Do abbreviations follow a simple primary rule?		

Table 12 Help users recognize, diagnose, and recover from errors checklist

#	Review checklist	Yes No N/A Comments
5.1	If toolbar can not verify the legitimacy of web pages, is user kept informed what user could do to check it manually?	
5.2	Is sound used to signal an error?	
5.3	Are popups brief and unambiguous?	
5.4	Are error messages worded so that the toolbar, not the user, takes the blame?	
5.5	Do prompts imply that the user is in control?	
5.6	Are error messages and warnings correct in grammar?	
5.7	Is wording in error messages and warnings in good manner or politely?	
5.8	Are warnings able to show the origin of error?	
5.9	Do warnings inform the user of the error's severity?	
5.10	Do warnings inform the user how to correct the error?	

Table 13 Error prevention checklist

#	Review checklist	Yes No N/A	Comments
6.1	Are menu items logical, distinctive, and mutually exclusive?		
6.2	Are data inputs case-blind whenever possible?		
6.3	Are data inputs type-sensitive?		
6.4	Does the toolbar alert users if they are about to make a potentially serious error?		
6.5	Do fields in data entry screens and dialog boxes contain default values when appropriate?		
6.6	Is there help or instruction for data inputs?		

Table 14 Recognition rather than recall checklist

#	Review checklist	Yes No N/A	Comments
7.1	For question and answer interfaces, are visual cues and white space used to distinguish questions, prompts, instructions, and user input?		
7.2	Are prompts, cues, and messages placed where the eye is likely to be looking on the screen?		
7.3	Have warning or error messages been formatted using white space, justification, and visual cues for easy scanning?		
7.4	Is it easy to find necessary operation for checking the legitimacy?		
7.5	Does the toolbar gray out or delete labels of currently inactive functions?		
7.6	Have items on a dialog or popup been grouped into logical zones, and have headings been used to distinguish between zones?		
7.7	Are significant button or menu item groups identified and highlighted?		
7.8	Does the toolbar provide mapping: that is, are the relationships between controls and actions apparent to the user?		
7.9	Are inactive menu items grayed out or omitted?		
7.10	Are the optional and non-optional settings of toolbar distinguished?		
7.11	Is it easy to find place for changing toolbar related settings?		

Table 15 Flexibility and minimalist design checklist

#	Review checklist	Yes No N/A	Comments
8.1	If toolbar supports both new and experienced users, are multiple levels of warning detail available?		
8.2	Can user customize the filter settings of toolbar?		
8.3	Can user customize the layout of toolbar?		
8.4	If menu lists are short (seven items or fewer), or there are limited buttons (no more than 10) on the toolbar, can users select an item or button by moving the cursor?		
8.5	Do users have the option of either clicking directly on a field (e.g. menu item, input field, and dialog box)or using a keyboard shortcut?		
8.6	Can users set their own default operation for the detected fraud?		
8.7	Can users nullify their default operation for the detected fraud?		
8.8	If users set their own default alert level too low, is there any warning or reminding for this?		

Table 16 Aesthetic and minimalist design checklist

#	Review checklist	Yes No N/A	Comments
9.1	Is only (and all) information essential to decision making displayed on the screen?		
9.2	Are meaningful groups separated properly?		
9.3	Does each group have meaningful title?		
9.4	Are menu items' titles brief, yet long enough to communicate?		
9.5	Do warnings concisely and correctly show the most important information about phishing detection?		
9.6	Can users be misled to other websites not related to phishing and its prevention?		

Table 17 Help and documentation checklist

#	Review checklist	Yes No N/A	Comments
10.1	Is there any user manual, tutorial, or help documentation available?		
10.2	Is there online help available?		
10.3	Is there instant help available?		
10.4	Is there necessary report to toolbar's provider available, when current version of toolbar can not successfully judge the web page?		
10.5	Do instructions or manuals follow the sequence of user actions?		
10.6	Are instructions and other help documentations understandable?		
10.7	Is it easy to search what user wants to know from help documentation?		
10.8	Is the help function visible and easy to find?		
10.9	Does the terminology of help documentations follow the toolbar general design conventions and standards.		
10.10	Is there context-sensitive help?		
10.11	Is it easy to access and return from the help system?		
10.12	Can users resume work where they left off after accessing help?		
10.13	Is the help detailed enough?		
10.14	Is there report to toolbar's provider available, when user can't find answer from existing help documentations?		

Table 18 Skills checklist

#	Review checklist	Yes No N/A	Comments
11.1	Can users learn from toolbar's functions or documentations what is phishing?		
11.2	Can users learn from toolbar's functions or documentations what are common phishing techniques?		
11.3	Can users learn from toolbar's functions or documentations how to identify the fraud WITH toolbar?		
11.4	Can users learn from toolbar's functions or documentations how to prevent the fraud WITHOUT toolbar?		
11.5	Can users learn from toolbar's functions or documentations how to protect their personal, or confidential information?		
11.6	Are there daily, or weekly phishing reports or news?		
11.7	Can users be informed about serious consequences, if users fail to follow the expected security advice?		

Table 19 Pleasurable and respectful interaction with the user checklist

#	Review checklist
12.1	Is each warning labeled properly in terms of its severity?
12.2	Can warning draw users' attention?
12.3	Is warning too overwhelming to disturb users' pleasant browsing?
12.4	Are there many false and undetermined detection warnings?
12.5	Are there any settings to simplify users' phishing detection sequences?
12.6	Are the frequently used function or button put in the most accessible position?

Table 20 Privacy checklist

#	Review checklist	Yes No N/A	Comments
13.1	Are protected areas completely inaccessible?		
13.2	Can protected or confidential areas be accessed with certain passwords?		
13.3	Is this feature effective and successful, or is the toolbar provider reliable enough to protect all of users' personal information submitted? (e.g. e-mail, telephone number, etc.)		

focused on the user's task, list concrete steps to be carried out, and not be too large (Table 17).

A.11 Skills

The toolbar should support, extend, supplement, or enhance the user's skills, background knowledge of anti-phishing —-not replace them (Table 18).

A.12 Pleasurable and respectful interaction with the user

The user's interactions with the toolbar should enhance the quality of her or his browsing experience. The user should be treated with respect. The design should be aesthetically pleasing—with artistic as well as functional value (Table 19).

A.13 Privacy

The toolbar should help the user protect any personal, private or sensitive information- belonging to the user (Table 20).

References

1. Anti-phishing working group (APWG): Phishing attack Trends Report—March 2006 (2006). http://www.antiphishing.org/reports/apwg_report_mar_06.pdf. Cited 9 Nov 2006
2. Chou, N., Ledesma, R., Teraguchi, Y., Boneh, D., Mitchell, J.C.: SpoofGuard (2004). http://crypto.stanford.edu/SpoofGuard/. Cited 27 July 2006
3. Downs, J., Holbrook, M., Cranor, L.: Decision strategies and susceptibility to phishing. In: Proceedings of the 2006 symposium On usable privacy and security, pp. 79–90 (2006)
4. Dhamija, R., Tygar, J.D., Hearst, M.: Why phishing works. In: The proceedings of the conference on human factors in computing systems (2006). http://people.deas.harvard.edu/∼rachna/papers/why_phishing_works.pdf. Cited 11 Nov 2006
5. Dinev, T.: Why spoofing is serious internet fraud. Commun. ACM, **49**(10), 76–82 (2006)
6. FBI National Press Office: Web 'Spoofing' Scams Are a Growing Problem. In: Press Release, Washington D.C. (2003) http://www.fbi.gov/pressrel/pressrel03/spoofing072103.htm. Cited 10 Nov 2006
7. Gartner Inc.: Gartner survey shows frequent data security lapses and increased cyber attacks damage consumer trust in online commerce (2005). http://www.gartner.com/press_releases/asset_129754_11.html Cited 22 November 2006
8. Google: Google safe browsing (2006). http://www.google.com/support/firefox/bin/static.py?page=features.html&v=2.0f. Cited 10 Oct 2006
9. Gutmann, P., Grigg, I.: Security usability. Secur. Priv. Mag. IEEE, **3**(4), 56–58 (2005)
10. Jakobsson, M.: Modeling and preventing phishing attacks. In: Phishing panel of financial cryptography (2005). http://www.informatics.indiana.edu/markus/papers/phishing_jakobsson.pdf. Cited 1 Nov 2006
11. Jakobsson, M., Ratkiewicz, J.: Designing ethical phishing experiments: a study of (ROT13) rOnl auction query features. In: Proceedings of the 15th annual World Wide Web conference, pp. 513–522 (2006)
12. Li, L., Helenius, M.: Anti-phishing IEPlug (2006). http://www.cs.uta.fi/∼ll79452/ap.html. Cited 1 Sep 2006
13. Netcraft: Netcraft anti-phishing toolbar (2006). http://toolbar.netcraft.com/. Cited 18 November 2006
14. Nielsen, J.: Heuristic evaluation online writings (1994). http://www.useit.com/papers/heuristic/. Cited 18 October 2006
15. Pierotti, D.: Usability techniques: heuristic evaluation—a system checklist (1998). http://www.stcsig.org/usability/topics/articles/he-checklist.html. Cited 18 October 2006
16. PhishTank: PhishTank—join the fight against phishing (2006). http://www.phishtank.com/. Cited 5 Nov 2006
17. Stop-phishing group (2006). http://www.indiana.edu/∼phishing/?people=external. Cited 20 Oct 2006
18. Wu, M., Miller, R., Garfinkel, S.: Do security toolbars actually prevent phishing attacks? In: Proceedings of the CHI 2006. 22–27 April 2006 Montréal, pp. 601–610 (2006)
19. Zhang, Y., Egelman, S., Cranor, L., Hong, J.: Phinding Phish: evaluating anti-phishing toolbars. In: Carnegie Mellon University, CyLab Technical Report. CMU-CyLab-06-018 (2006). http://www.cylab.cmu.edu/default.aspx?id=2255. Cited 15 Nov 2006

Study 5

Li L., Berki E., Helenius M., Ovaska S., (2012). Towards a contingency approach with whitelist- and blacklist-based anti-phishing applications: what do usability tests indicate? Submitted to: Behaviour & Information Technology Journal

.

Towards a contingency approach with whitelist- and blacklist-based anti-phishing applications: what do usability tests indicate?

Linfeng Li

School of Information Sciences, University of Tampere

Tampere, Finland

Linfeng.li@uta.fi

Eleni Berki

School of Information Sciences, University of Tampere

Tampere, Finland

Eleni.Berki@uta.fi

Marko Helenius

Department of Pervasive Computing, Tampere University of Technology

Tampere, Finland

marko.t.helenius@tut.fi

Saila Ovaska

School of Information Sciences, University of Tampere

Tampere, Finland

Saila.Ovaska*@uta.fi*

Towards a contingency approach with whitelist- and blacklist-based anti-phishing applications: what do usability tests indicate?

Abstract

In web browsers, a variety of anti-phishing tools and technologies are available to assist users to identify phishing attempts and potentially harmful pages. Such anti-phishing tools and technologies provide Internet users with essential information such as warnings of spoofed pages. To determine how well users are able to recognize and identify phishing web pages with anti-phishing tools, we designed and conducted usability tests for two types of phishing-detection applications: blacklist-based and whitelist-based anti-phishing toolbars. The research results mainly indicate no significant performance differences between the application types. We also observed that, in many web browsing cases, a significant amount of useful and practical information for users is absent, such as information explaining professional web page security certificates. Such certificates are crucial in ensuring user privacy and protection. We also found other deficiencies in web identities in web pages and web browsers that present challenges to the design of anti-phishing toolbars. These challenges will require more professional, illustrative, instructional, and reliable information for users to facilitate user verification of the authenticity of web pages and their content.

Keywords: phishing; phishing prevention; contingency; privacy; human behaviour; social engineering

Subject classification codes: H.5.2 [User Interfaces], K.6.5 [Security and Protection]

1. Introduction

Internet phishing is a typical form of identity theft whereby the main aim is to obtain confidential information such as credit card numbers, personal credentials, and social security ID numbers. Many phishing and spam attempts disrupt people's lives (Li *et al.* 2011a), particularly when the information obtained is later used to carry out malicious activities. In the year 2011 alone, thousands of phishing pages have been created (APWG 2011).

Various approaches and technologies have been proposed and developed (Li *et al.* 2012b) to protect people from phishing, with varying degrees of success (Li *et al.* 2011b, Li *et al.* 2012a). One promising and ongoing development in the phishing-prevention field is that of anti-phishing client-side applications such as toolbars or plug-ins for web browsers. In general, these anti-phishing toolbars and plug-ins are able to detect forged web pages and warn Internet users about suspicious web page content while (or as soon as) the content loads. From a user security point of view, the most significant advantage of anti-phishing toolbars is that because of their anti-phishing technology they are able to stop phishing attempts before confidential information is given. Nevertheless, the performance of existing anti-phishing applications is not satisfactory (Wu *et al.* 2006a). Technical challenges include the frequency of updates to anti-phishing blacklist databases, and false negatives in self-adaptive detections (e.g. some phishing pages cannot be detected by self-adaptive algorithms because the algorithms use inadequate training or corpora collections (Li *et al.* 2011b)). These are only two of the many technical obstacles. If anti-phishing blacklist databases are not frequently updated, users may not be fully informed and can consequently be misled into visiting phishing web pages. This can easily become a common occurrence if the non-updated anti-phishing toolbars fail to warn the user of the urgent need for an update. Phishing web pages may also bypass, with no warning, toolbars that use adaptive content filtering. Two reasons for this may be the toolbars' user-friendliness and ease-of-use; these are rarely emphasized in the phishing-prevention context. Usability issues in general are not paid the same attention as security and confidentiality aspects in the design of anti-phishing toolbars. Furthermore, phishing-prevention applications that are user-centred remain under-designed and under-utilized, and there is room for significant improvements through future research and development (Hong 2012).

Although a wide variety of anti-phishing toolbars are available, phishing and phishers in general are becoming increasingly sophisticated (Li *et al.* 2012b). Moreover, social engineering is taking on new forms that are exploiting people's everyday lives (Dimensional Research 2011). The millions of web pages making up the Internet are part of an immense social network that is influenced daily by the actions of millions of people worldwide. In that respect, web researchers and developers should be aware of people's cognitive needs and mental models because these factors can aid design work. A basic understanding of social-engineering concepts can also be valuable to technical designers of anti-phishing and anti-spam technologies that affect people's social interactions.

For the above main reasons, we decided to explore the capabilities of anti-phishing toolbars and their detection mechanisms, with the aim of determining recognition features and adapting them to create an effective design approach that utilizes user experiences. This study in particular obtained findings that may be useful both to researchers into web spam and anti-phishing technology and to cognitive and user psychologists.

To simplify our research questions, we considered two detection mechanisms that represent two types of anti-phishing applications:

- Blacklist-based anti-phishing applications: warn a user when the URL of the visited web page is on a list of already detected and/or reported phishing web pages.
- Whitelist-based anti-phishing applications: confirm the authenticity of trustworthy web pages that have been saved to a whitelist. If the domain name of the web page being opened is found to be similar to a domain name on the whitelist, the application generates a warning message to allow users to determine its authenticity.

For our usability test, our assumption was that *a blacklist-based anti-phishing toolbar would be able to help users identify more phishing pages than a whitelist-based one*. We conducted a usability test that was designed so that i) this hypothesis could be tested and ii) further specific comments from the participants could be collected. Although we found no significant performance differences between the blacklist-based and whitelist-based applications, we made a number of other interesting findings. One notable observation we made was that current anti-phishing mechanisms of web browsers are not very effective. They lack considerable information regarding user security, and this alone can mislead users into visiting phishing web pages and/or providing personal information to phishers.

The remainder of this paper is structured as follows. In Chapter 2 related work in the anti-phishing research area is discussed. In Chapter 3 the methodology including the test setup is presented. In Chapter 4 we present our results, and in Chapter 5 we analyse and discuss the most significant findings together with participants' feedback.

2. Anti-phishing toolbars: related research and development work

A variety of scientific research has attempted to establish why phishing attacks are so successful; some of these studies are briefly mentioned in this section. For example, Jakobsson and Ratkiewicz (2006) estimated how successful phishing attacks can be in practice. In their study, they spoofed 'blinded' participants through social phishing. The participants were unaware of the design of the study. The researchers created a virtual phishing environment and conclude that the success rate for a sub-domain link attack is even higher than that reported in a study by Linta (2004), who found that 19% of the victims clicked on phishing links in their emails. Dhamija *et al.* (2006) attempted to establish problems in users' browsing behaviour, e.g. neglecting to check domain names. Blythe and co-workers (2011) also attempted to discover why phishing is often so persuasive.

Analysing the above-mentioned and other characteristics of phishing attempts in the research literature, Blythe *et al.* (2011) suggest building a reading strategy for end users to identify phishing attempts effectively. The researchers also conducted a web survey asking participants to identify phishing emails. The participants were asked to identify i) phishing emails and ii) genuine emails from a set of collected email samples with the participants' own computers. They returned their answers and feedback through an online survey system. Blythe *et al.* (2011) found that the survey participants detected fewer phishing web pages with (commercial) logos than without logos. However, the environment was not controlled. Hence, the results of the survey lacked reliability because of the lack of control over the security configurations of the participants' systems and web browsers.

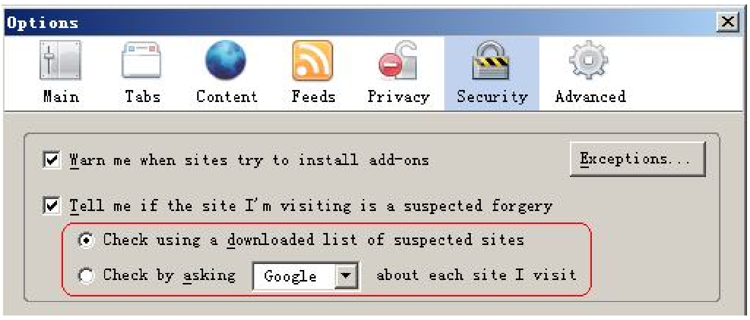
Some usability evaluation research studies have attempted to improve anti-phishing toolbars in web browsers. Zhang *et al.* (2007) carried out a performance evaluation of five popular anti-phishing toolbars and revealed their limitations. Although the accuracy of the results of this type of evaluation is doubtful, they clearly show that the performance of the tested toolbars was not satisfactory. In addition, Egelman *et al.* (2008) collected and studied passive and active warnings provided by the Firefox 2.0 and Internet Explorer 7 web browsers. They found the most effective warning to be an active one that prevents users from visiting a suspicious web page by actively displaying a blocking page. Egelman *et al.* (2008) also state that warnings in the phishing context should be able to interrupt users' primary tasks and to offer appropriate recommendations on safe web browsing. However, these studies (Zhang *et al.* 2007, Egelman *et al.* 2008) did not go on to evaluate usability between blacklist-based and whitelist-based anti-phishing applications.

A number of anti-phishing applications and tools have been designed, developed, and evaluated from a usability perspective. For example, Luca *et al.* (2011) observed the experiences of users of a novel security human–computer interface, the MoodyBoard, which is a keyboard with an enhanced security feature to notify end users. Luca *et al.* (2011) demonstrated that MoodyBoard's ambient notifications have a positive influence on secure behaviour. However, in their experiment the researchers did not consider the effects of web browsers' security warnings. This is of importance, particularly when web browsers today have inbuilt anti-phishing features. Moreover, Lin and his colleagues (2011) investigated one of these default anti-phishing features in Internet Explorer, called 'domain highlighting'. They found that domain highlighting is ineffective from the usability point of view; in fact, almost two-thirds of fraudulent pages were rated as safe. The researchers suggest placing more emphasis on user education, user awareness, and URL complexity.

Concerning user education, Villamarín-Salomón and Brustoloni (2010) investigated user security-behaviour training and carried out a user study on whether security-reinforcement applications assist users in improving their security behaviour.

By following an improvement policy based on social learning theory (Bandura 1977), the learning effects of security-reinforcement applications can be accelerated by explicit security reinforcement. Sheng *et al.* (2010) also performed an anti-phishing experiment from the perspective of user education and training. They found that anti-phishing training materials can assist in educating users to identify phishing attempts. However, the researchers did not explore the learning effects of users' interactions with anti-phishing tools.

In our research, we used Google Safe Browsing and Anti-phishing IEPlug (2006) as representative blacklist and whitelist-based anti-phishing applications, respectively. Google Safe Browsing is a toolbar included in the Firefox web browser (2007). It is a typical anti-phishing application based on blacklist detection. Google Safe Browsing inspects any visited link. At the time of our study, it offered two detection options (see Figure 1). The first option uses a downloaded list of suspected sites (blacklist), and the other option uses a blacklist stored on Google's server. In our test, we selected the first option. According to related evaluations (Zhang *et al.* 2007, Egelman 2008, Li and Helenius 2007, Li *et al.* 2012a), Google Safe Browsing has remarkably good detection rates and provides a number of sufficient warnings; these evaluation results alone made Google Safe Browsing a good candidate for inclusion in our usability test.



☑ Warn me when sites try to install add-ons

☑ Tell me if the site I'm visiting is a suspected forgery

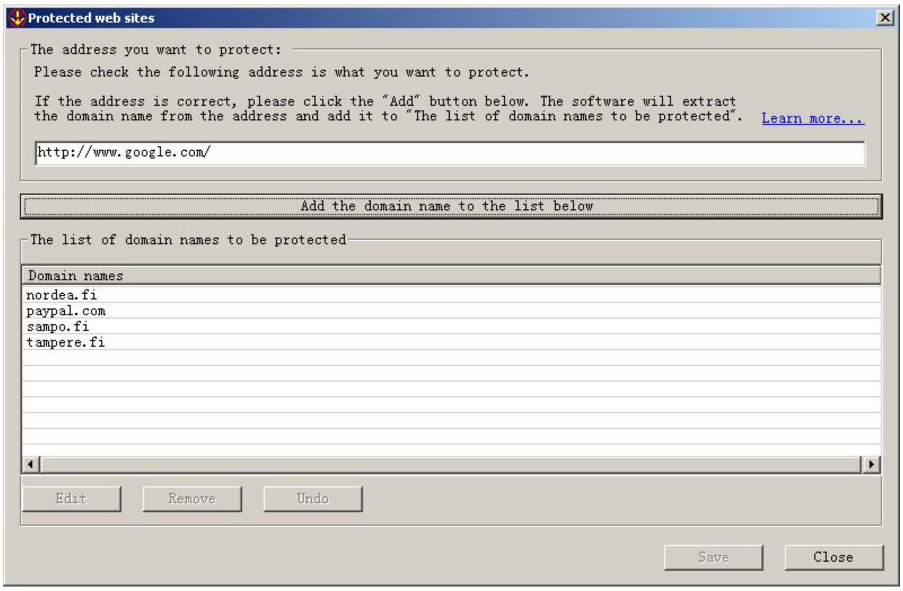
● Check using a downloaded list of suspected sites

○ Check by asking Google about each site I visit

Figure 1. The detection options of Google Safe Browsing are circled.

Web Wallet, another anti-phishing application, is able to detect phishing attempts based on a whitelist. As such, Web Wallet can prevent a user from providing a password to unauthorized or unrecognized websites (Wu *et al.* 2006b). However, the key feature of Web Wallet is the management of users' private online information.

When we started our research, no whitelist-based anti-phishing applications were available, including Web Wallet. Therefore, we decided to develop a novel whitelist-based anti-phishing toolbar, Anti-phishing IEPlug (2006), hereafter called IEPlug (see Figure 2).



Protected web sites

The address you want to protect:

Please check the following address is what you want to protect.

If the address is correct, please click the "Add" button below. The software will extract the domain name from the address and add it to "The list of domain names to be protected". Learn more...

http://www.google.com/

The list of domain names to be protected

Domain names
nordea.fi
paypal.com
sampo.fi
tampere.fi

Figure 2. The main GUI of Anti-phishing IEPlug.

After a user has added safe domain names to the IEPlug whitelist, the application inspects web pages the user visits. IEPlug provides a certificate authority (CA) dialog box for safe web pages, and warns the user when a fraudulent page is detected. If a password text field is on the web page and the domain is whitelisted, the application displays information in the CA dialog box for that page.

The appearance of the certificate makes it obligatory for the user to pay attention to the essential security features of the web page being visited. Letting the user know what should be checked and verified before any online (e.g. banking) transactions are conducted can be an effective educational technique.

Visual dialog windows may, of course, disturb users' workflow. As a pilot test application, however, this experimental procedure helped us to examine our assumption, which is whether whitelist-based anti-phishing tools with informative and straightforward certificates help users to identify more online fraud.

Both Google Safe Browsing and IEPlug have limited capabilities in terms of warning about suspicious and/or fraudulent web pages. Furthermore, the blacklist information may be incomplete, and consequently, the browser may display no warnings. If the domain of the visited URL is not whitelisted but the URL contains one of the domain names stored on the whitelist, the program will halt the user's browsing and alert the user to possible web page forgery. IEPlug warns the user about an empty whitelist to allow users to familiarize themselves with IEPlug, in order to reduce the likelihood of later misuse of the application. For example, if the whitelist were to remain empty, the user would never see a CA dialog box.

3. Research methodology and research procedures

Usability testing (Nielsen 1993) is a research methodology for collecting information on users' behaviours and preferences within a prepared and experimental (therefore controlled) environment. During the test, the test object (software) and a set of pre-defined test tasks are given to the invited participants. By observing and analysing the particular behaviour and participants' feedback, the researcher is able to determine from real users specific requirements to help application designers to improve their designs. To avoid any bias, the researchers must not interfere with the test, but simply observe the procedure and collect the required data for further analysis.

To test our hypothesis in the same phishing attack context for each participant, we controlled the participants' browsing behaviour as well as the order and the type of web pages tested. In the following sections, we describe the details of the design and representative parts of the usability test.

3.1. Phishing pages simulation

To make the tests comparable, the number of phishing pages detected was the same in the test sessions of both applications. We collected phishing pages detected by Google Safe Browsing from the anti-phishing community site PhishTank (2011). These pages were verified to make sure that Google Safe Browsing would be able to detect them during our usability test.

For the test, we counterfeited the domain names of real and invented online financial services with authentication mechanisms in four ways:

- Domain name with visual similarity: resembles an authentic name but with some characters altered. For example, we forged paypal.com as paypol.com;
- Domain name with semantic similarity: uses strong semantic implication, such as securelogin.com;
- Domain name with the same sub-domain name: The sub-domain name (normally the first part of the URL) is indistinguishable from the domain name of a real online service. For instance, we forged the phishing link for signin.ebay.com/ws/eBayISAPI.dll?SignIn as signin.ebay.com.ws.eBayISAPI.com;

- Identical domain name resembling a DNS (Domain Name Server) hijacking attack: Domain names and IP addresses in the DNS server are manipulated, and DNS requests may be misdirected to malicious websites.

Many pages were forged using more than one of the above methods. For example, one fake URL was www.login-paypol.com/secure-register, where both visual similarity and semantic similarity were employed.

3.2. Preparation and design of the usability test

The whole experiment was conducted in a controlled environment. We selected the six most popular online web services and collected thirteen test web pages in each of the two test sessions. To prevent external access to these pages, we established our own isolated local network containing customized DNS and WWW (World Wide Web) servers that were under our continuous control.

3.2.1. Invited participants

For our research study, twenty student participants from the university campus were invited. There were four females and sixteen males aged 20–50 years. Two of them were native English speakers, and the remainder were able to fluently communicate in English during the test and when completing the test tasks. All of the participants reported that they used online banking services frequently and that every week they spent at least five hours using the Internet. The test participants were voluntarily involved in the testing procedure and were given no reward, including money, for their participation in the test groups.

Prior to the test, we collected information on participants' online behaviour through the following questions:

1. How many hours a week are you online?
2. What do you do on the Internet?
3. How long have you been using online payment?
4. How much do you know about domain names in general?
5. Do you know about SSL (Secure Socket Layer)/TLS(Transport Layer Security)?
6. Can you explain each step in an online payment?

In general, the invited participants in the two groups (ten participants for each toolbar application) had similar limited online experiences and IT knowledge. The average weekly time online was 25 hours (Google Safe Browsing) and 35.4 hours (IEPlug), and there was no significant difference after a two-tail t-test (t=1.038, p=0.313). Average online payment experience was 3.3 years (Google Safe Browsing) and 2.7 years (IEPlug), and we found no significant difference between them after a two-tail t-test (t=0.851, p=0.407). Statistical details are listed in Table 1. Other online activities of the participants were reading online news and emails, and e-learning. Two out of the ten (Google Safe Browsing) and none out of the ten (IEPlug) participants had some knowledge of SSL. Three out of ten (Google Safe Browsing) and one out of ten (IEPlug) participants knew what a domain name was and were able to describe each part from the given URL. One out of ten (Google Safe Browsing) and none of the ten (IEPlug) participants had focused on security and privacy issues when dealing with online payments.

Table 1. Participant statistics.

	How many hours a week are you online?		How long have you been using online payment?	
	Mean	Standard Deviation	Mean	Standard Deviation
Google Safe Browsing participants	25 hours	18.7735 hours	3.3 years	1.337 years
IEPlug participants	35.4 hours	25.5265 hours	2.7 years	1.783 years

3.2.2. Usability test tasks

In our research study, we explicitly asked the participants to check and report on the authenticity of the given web pages. Checking the authenticity of the given web pages was the *main* test task. We also designed other test tasks to allow the participants to familiarize themselves with the functions of the tested applications. The participants who tested Google Safe Browsing were requested to confirm that Google Safe Browsing uses a downloaded list of suspected websites. Those who tested IEPlug were asked to add a domain name to the whitelist and to state what else had been added to the whitelist.

3.2.3. Training tutorials

The training tutorials for the two applications to be tested were not available in the same way. The tutorial for Google Safe Browsing (2007a) was available only when a user downloaded the tutorial or when a warning of web forgery appeared. IEPlug forced its users to read the related tutorial; there was easy access to the tutorial through the whitelist configuration dialog. For further testing of our hypothesis, it was necessary to balance users' learning, by requesting the participants to undertake additional reading from a separate paper-based tutorial. This took place before starting the actual usability test for both applications and both groups of participants.

These official tutorials were printed on paper and given to the participants at the beginning of the test. For IEPlug, we printed its tutorial. For Google Safe Browsing, we collected the information from two web pages. One was the Toolbar for Firefox Help Centre, where Safe Browsing was introduced as a feature of the Google toolbar for Firefox (2007b). The other page was the Phishing Protection page (2007) at the Mozilla website. This page is shown when a user clicks the links *'Read more…'* and *'How Firefox protects you'* on the web forgery warning.

3.2.4. Network topology

The network topology we used is illustrated in Figure 3. The computer used by the participants was directly connected to a DNS and web server. The DNS and web server were used to resolve web addresses and to reveal phishing websites. By using the DNS server, we could control any web page requests and, thus, show fake pages with any addresses as necessary. Requests for authentic pages were forwarded to the DNS server outside our controlled environment. For security reasons, if any requests came from the external network outside the computer laboratory, the firewall rejected them.

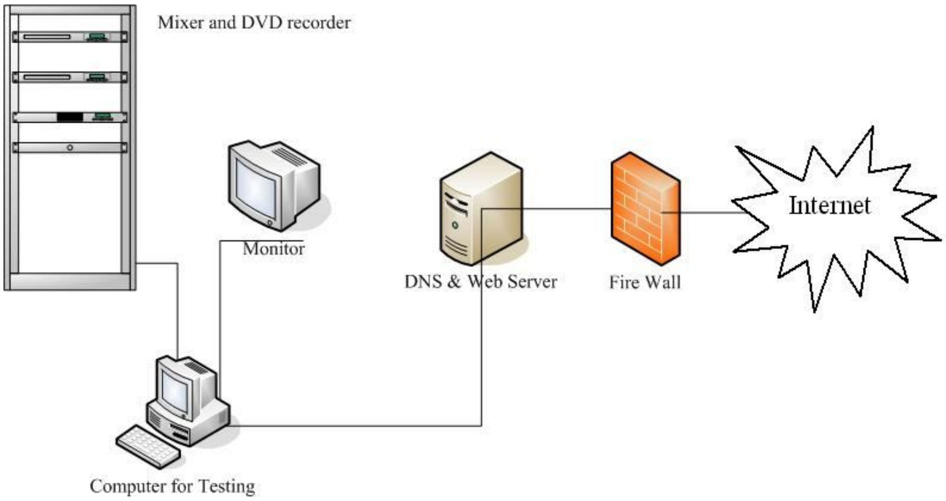


Figure 3. The network topology for the usability tests.

3.2.5. The test procedure: pre-design and test realization

We used the following procedure to control each test session. The complete test comprised one tutorial session, one interim interview, and two test sessions (Figure 4). Each participant was asked to test only one of the selected anti-phishing toolbars. All participants attended the same test sessions. In general, the duration of one complete test procedure was approximately two hours for each participant.

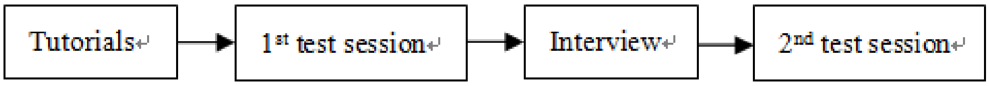


Figure 4. The test procedure.

For the first test session, a sheet of paper containing the tutorial for the application was first shown to the participant. After the participant had read the tutorial, we showed the pool page that contained anonymous links to web pages in a pre-specified order. The pre-specified order was the same for all participants. The participants were required to visit each test page and, at the same time, were asked to think aloud as they decided on the authenticity of the visited web page. If they did not, participants were asked whether the visited web page was authentic (participants were required to answer ‘Yes’ or ‘No’) and were asked to give reasons.

The pool pages included thirteen links of six popular e-service websites that are commonly used in Finland. These were three banks (Nordea, Sampo bank, and Osuuspankki), Tampere City library, eBay, and PayPal. In the first test session, two phishing web pages were correctly detected by the toolbar, five phishing web pages were undetected, and six pages were authorized as authentic. In the second test session, one phishing page was detected, eight phishing pages were undetected, and four pages were authorized as authentic. Afterwards, the participants were asked to state whether or not they trusted these pages and, subsequently, were allowed to use any method to determine their authenticity, including using the anti-phishing application being tested and/or other manual methods.

Those phishing pages that we did not want Google Safe Browsing to detect were locally created and stored and, thus, never stored on the blacklist. As IEPlug’s phishing-detection mechanism is based on analysis of keywords (i.e. the domain names of the pre-saved whitelisted pages), we simply controlled the appearance of a keyword in the domain name. In doing so, we were able to test how many web pages could correctly be identified by users with the anti-phishing toolbar being tested.

Even though the tutorial sheets given at the beginning of the first test session assisted in mimicking learning, the quality of these tutorials and users’ understanding naturally varied. Therefore, before the second test session, we carried out an interim interview to determine how well participants had learned from the tutorials and their usage of the tool being tested. If a participant was not yet acquainted with some part of the information necessary for detecting fraud and phishing, we offered further details. In this way, we aimed to ensure that every participant knew everything that they needed to in order to use the tested application to prevent phishing attacks.

After the interim interviews and subsequent instructions, we asked the participants to visit another thirteen web pages on the second pool page. This time we used a greater number of phishing web pages (nine). Among these phishing pages there was only one correctly detected page; the applications failed to detect the remaining web pages.

3.2.6. Data collection: techniques and methods

Our collected data from the usability test were gathered with mainly two methods. In the first method, we recorded the interactions in the laboratory during the usability test. These interactions included the users' input (from mouse and keyboard), audio stream, and content displayed on the computer monitor. In the second method, we asked participants to fill in a structured questionnaire to collect feedback regarding the anti-phishing toolbar being tested. The questionnaire mainly focused on gathering data about the following: i) general feelings about the particular anti-phishing toolbar; ii) user experiences concerning toolbar warnings; iii) learnability and reliability of the anti-phishing toolbar. Participants rated their attitudes on a five-point Likert scale (1: disagree the most; 5: agree the most).

4. Research study results and discussion

After the twenty usability tests, we analysed the videos and the answers to the questionnaire. In the following sections we provide insights and comments by comparing and contrasting the findings from the two participant groups. First, we reviewed all the videos once, and then noted the participants' choices and the reasons for these choices. The key activities during these reasoned choices were also documented.

4.1. Results reported from the test sessions

The key activities during the usability test provided us with plenty of information regarding the participants' behaviour. More specifically, we felt that when the participants attempted to justify their choices and decisions on the authenticity of the web pages, this was the most valuable part of our research study because it provided us with the most essential data. We next outline and analyse specific details of interest that are within our research scope.

4.1.1. Accuracy rate of page identification

To better compare the test results for the two applications, we calculated the accuracy rate. By accuracy rate we mean the number of correct answers that participants provided during all tasks in a test session. For a respondent's reply to be considered a correct answer, two conditions must be met, which can be summarized as the correct decision associated with the correct reasons. Figure 5 shows that the average accuracy rate was more than 60%; however, a huge distribution within groups was found. In addition, participants correctly identified more phishing web pages in the second test session than they did in the first test session. This improvement is statistically significant for both tested objects (Google: $t = 2.508$, (two-tailed) $p=0.033$; IEPlug: $t= 3.483$, (two-tailed) $p=0.007$). The means and the standard deviations are listed in Table 2.

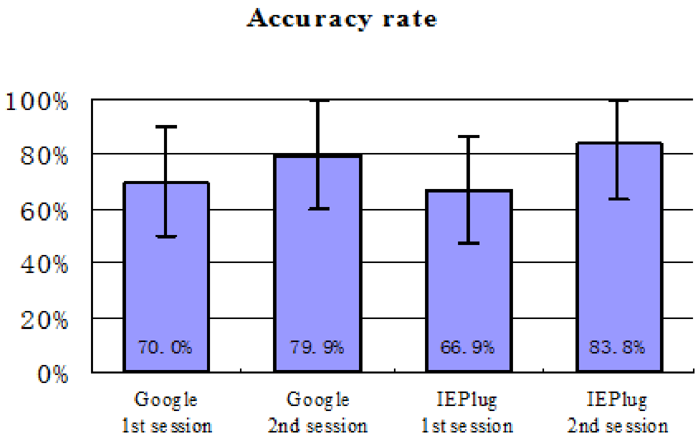


Figure 5. Accuracy rate and standard deviation for each test set.

Table 2. The mean and standard deviation of accuracy rate for each test set.

	Google Safe Browsing, 1st session	**Google Safe Browsing, 2nd session**	**IEPlug, 1st session**	**IEPlug, 2nd session**
Mean	70%	79.9%	66.9%	83.8%
Standard Deviation	26.76%	29.71%	30.35%	21.26%

This result shows that relevant learning tutorials do help users to find a greater number of suspicious web pages. They also support the research findings of Wu *et al.* (2006a), Villamarín-Salomón and Brustoloni (2010), and Sheng *et al.* (2010). Wu *et al.* (2006a) in particular found that specific tutorials result in learning effectiveness relating to spoofing rates. Villamarín-Salomón and Brustoloni (2010) concluded that following a security policy by building a user-behaviour model and providing education on personal security matters are more effective than security-reinforcement applications. Sheng *et al.* (2010) found that sufficient phishing education can assist users in correctly identifying phishing attempts.

Comparing these results with our research findings, we can only affirm that the tutorials in our usability test may not have been effective in educating users to correctly identify phishing web pages with the tested anti-phishing toolbars. The content and/or the method of training may need improvement.

Although the accuracy rates between Google Safe Browsing and IEPlug varied, the difference is not significant (comparison between 1st sessions: t =0.240, (two-tailed) p=0.813; comparison between 2nd sessions: t =0.339, (two-tailed) p=0.739). This non-significant difference implies that explicitly presenting the CA dialog box may not lead users to make informed decisions. A similar result was also obtained by Lin *et al.* (2011).

4.1.2. Understanding the domain name

The domain name is a strong indication of a reliable web page (see, for example, Microsoft 2006). It has certain cognitive associations with other physical and/or abstract domains. Furthermore, a simple and effective way for users to protect themselves against phishing is to always type in the domain name of a critical service. Before the interim interview, seven (out of ten) participants using Google Safe Browsing and seven (out of ten) participants using IEPlug deliberately checked the domain name during the test. However, after they had learned more about how the software worked, eight Google Safe Browsing participants and nine participants using IEPlug learned (that it was important) to also check the domain name.

An interesting observation was that, even though every participant was able to successfully extract the domain name during the first interview, only three participants used the domain name to identify the authenticity of the web page during the first testing session. Moreover, three participants (one using IEPlug, two using Google Safe

Browsing) concentrated on less important parts of the URL, and the interim interview had no effect on how they interpreted URLs. For example, when the address *https://signin.ebay.com/ws/eBayISAPI.dll?SignIn* appeared, some participants admitted that they had never before seen a web page that used a .dll file. An even more alarming finding is that eight participants out of the twenty focused only on the beginning of the link. For example, when these participants – four using Google Safe Browsing and four using IEPlug – inspected the address *https://signin.e-bay.com.ws.eBaISAPI.com/index.html*, the only part that was noticed was 'www.e-bay.com' and, thus, they missed the actual domain: eBaISAPI.com. These test participants appeared to be ideal victims for phishers.

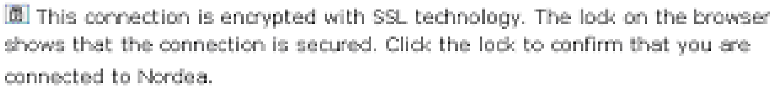
In the first test session for Google Safe Browsing, only four out of ten participants used the domain name to identify the authenticity of the web page. After emphasizing and clarifying the interim question(s), four more participants were able to identify phishing web pages by checking the web address, although sometimes they did not check the correct part of the domain name.

Although IEPlug explicitly displayed the correct, safe domain name in its warning, participants still neglected to check the domain name of the visited page or even the entire web address. Interestingly, two participants unexpectedly exploited a specific capability of the software to find out the domain name of the web pages tested. They achieved this by adding the URLs of the visited pages to the whitelist, and because of IEPlug's capability of extracting the actual domain name from a given URL, it added only the domain part to the whitelist. For example, when one user added the long URL *https://solo1.nordea.fi.nsp.engine.space.securelink.onlinebank.com* to the whitelist, only 'onlinebank.com' was added to the whitelist, and this assisted the user in identifying the visited web page.

4.1.3. Secure Socket Layer awareness

Another reliable method of identifying web pages is the CA for web servers. In our study, checking the CA appeared to require more technical proficiency than the domain name. In fact, for most participants the CA was a mystifying and hidden prompt. Sixteen out of the twenty participants had insufficient knowledge about SSL or TLS protection and, thus, did not know how to verify the security certificate. Hence, most of the participants were prone to sophisticated phishing attacks.

Only one Google Safe Browsing participant correctly identified phishing pages by checking for secure connection indicators in the first testing session. In the second testing session, the number of participants who did this increased to three. The two new participants learned how to check the lock icon from the instructions on the tested web pages (see Figure 6).



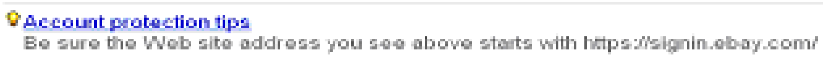
🔒 This connection is encrypted with SSL technology. The lock on the browser shows that the connection is secured. Click the lock to confirm that you are connected to Nordea.

Figure 6. The secure connection instruction on the web page of Nordea bank.

Because IEPlug was able to show the CAs of safe websites, all of the IEPlug test group participants knew that *something* should be displayed when they visited an authentic page. However, displaying the CA alone does not guarantee safe browsing. First, the content of the CA is too technical for users to fully understand. Even though the participants were asked to read the IEPlug tutorial, it contained no information about how to interpret CAs. Furthermore, IEPlug only displayed the CAs and did not inform users how to check them. Therefore, four out of ten IEPlug participants misidentified one authentic web page as a phishing page, which was in fact intentionally omitted from the whitelist. The participants' reasons for this misidentification were the same: *'There is no certificate shown. According to the tutorial, I should not trust it.'* This indicates that some key conditions needed to operate correctly the whitelist-based application were ignored by users.

4.1.4. Security hints on web pages

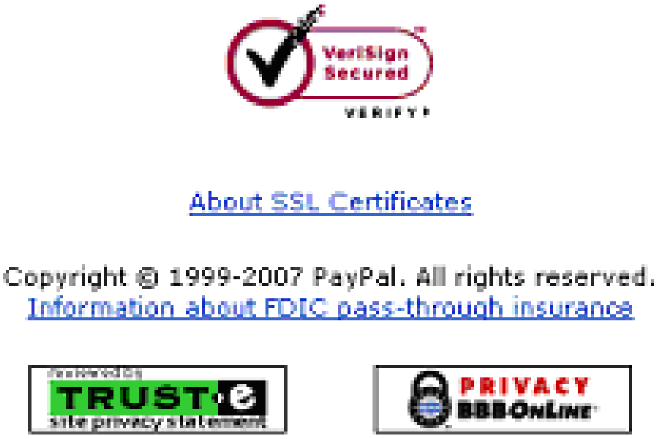
Besides standard forms of identification such as domain names, some security hints or instructions may also be provided on web pages, including domain name instructions (Figure 7), secure connection instructions (Figure 6), and security verification icons by third-party organizations (Figure 8). These hints are helpful to users who are unsure whether they should trust a service.



Account protection tips

Be sure the Web site address you see above starts with https://signin.ebay.com/

Figure 7. Domain name notifier.



About SSL Certificates

Copyright © 1999-2007 PayPal. All rights reserved.
Information about FDIC pass-through insurance

Figure 8. Security verification icons.

However, these hints alone are not sufficient for user protection. This is because it is easy to abuse and/or change these hints. The hints are usually in the form of text or links on the web pages, and phishers can easily alter them into something harmful to visitors. For example, the source code for the Verisign icon can effortlessly be modified in the following way:

```
<a href=”http://www.evilsign.com/ebay.com.html”>
<img src=”verisign.gif”></img></a>
```

Subsequently, when a user trusts the icon and clicks on it, the verification information displayed on the page opened by the forged link is identical to the authentic version. Finally, some of these security hints are ambiguous and sometimes confusing. For example, in the instruction in Figure 6, the user is simply asked to ‘click the lock’, which misled one of our participants into clicking the lock icon on the web page. However, the correct place to ‘click the lock’ is at the address bar or the status bar of web browsers.

4.1.5. Regular elements on web pages

One interesting finding is that seven out of the ten Google Safe Browsing participants and four out of the ten IEPlug participants preferred to check layouts or some insignificant functions on web pages during the first test session. For example, the method most used was to try the links on the page. In the first test session, seven Google Safe Browser participants and five IEPlug participants clicked at least one link to check the authenticity of the web page. Obviously, this alone does not protect users against phishing attempts, particularly when phishers can frequently access and modify every link on a phishing web page. Moreover, the favourite links to click were ‘help’ or ‘privacy policy’, where there may simply be content to which phishers can very convincingly point the user.

In the second session, even more participants focused on these links of the tested web pages (nine participants testing Google Safe Browsing and four participants testing IEPlug). Interestingly, the participants focused more on the content of the web pages after the interim interview had emphasized how to use the application correctly. It appears that IEPlug enabled users to use the relevant forms of web page identification better than they did through Google Safe Browsing. Therefore, an IEPlug user knew what to trust and what not to trust. For example, IEPlug was able to i) explicitly confirm the authentic domain names that were stored on the whitelist and ii) displayed CAs as supplementary information or evidence-based knowledge.

4.1.6. Spoof awareness

Amazingly, two participants (or three, if we count in one of the three pilot test participants) did not fully understand what an authentic page was, even throughout both test sessions. They appeared to believe that any web page requesting a user account and password was an authentic page. This reveals an important deficiency in the average person's mental model: these participants were used to proving their own identities, but neglected to verify the trustworthiness of online service(s). For instance, in our test, we found that two participants (one from the IEPlug group and one from the Google Safe Browsing group) always followed these criteria to decide on the authenticity of web pages. In reality, they trusted the web pages as long as the service required confidential information to log in. Furthermore, they even overlooked, underestimated, or did not trust the warnings from the two applications being tested, and followed their own rules instead.

All in all, this is an astonishing finding: from a total of twenty participants, we found that, among this group, three participants were ideal victims even for simple and unsophisticated phishing attacks in which several one-time passwords are requested. With our data set as reference, this particular finding could be almost as frequent as every seventh user.

4.1.7. Warnings

The warning system is a key component of a client-side anti-phishing application, and should be capable of stopping users from visiting suspicious websites. However, during the pilot test of IEPlug, one participant simply clicked the default button on the warning dialogs without reading through the content, even when a special warning popped up. Fortunately, the application provided a second warning, which worked well in our test case: it stopped all participants making a dangerous or faulty decision, even though not all participants fully understood or read the warning message.

4.1.8. Trusting the software

Participants did not find it easy to understand how the applications worked. Without proper understanding, they easily formed misconceptions and wrong opinions. For example, some participants over-relied on what they were using, particularly when they did not fully understand how the application worked. However, the seemingly limited performance, that is the limited number of phishing pages detected, caused twelve of the participants (eight from the Google Safe Browsing group and four from the IEPlug group) to be so disappointed that they stopped relying on and trusting the software.

4.2. Results from the questionnaire

After the test sessions, all participants filled in the questionnaire (Table 3). The results are shown in Figure 9 and Figure 10. From participants' answers, both tested toolbars appeared to be easy to use.

Table 3. The questionnaire.

Statements	***Answers***
It is easy to use.	I do not know□, Disagree 1□ 2□ 3□ 4□ 5□Agree
It is disturbing my browsing.	I do not know□, Disagree 1□ 2□ 3□ 4□ 5□Agree
Its warnings are easy to understand.	I do not know□, Disagree 1□ 2□ 3□ 4□ 5□Agree
Its warnings can catch my attention.	I do not know□, Disagree 1□ 2□ 3□ 4□ 5□Agree
I can easily learn what is phishing (or spoofing) by using it.	I do not know□, Disagree 1□ 2□ 3□ 4□ 5□Agree
I can rely on it for identifying spoofed web pages.	I do not know□, Disagree 1□ 2□ 3□ 4□ 5□Agree

The evaluation results of the warnings were similar for both applications, but the results also indicated that the warnings of the Google Safe Browsing were preferred. The main concern of the participants was the (perhaps frequent) popping up of CAs. Seven IEPlug group participants indicated that this was annoying when they were visiting web pages. One, somewhat experienced, participant wrote: *'It seems to disturb a bit, since it opens the cert. And so on. I do not need to see the cert. I can open it myself.'* One inexperienced participant wrote: *'a little bit, the cert. form is ok, but another prompt form seems to give me something. Important information, so when users first saw it, they must try to understand what's going on.'*

When the participants were requested to consider the learning effects of these two applications, they gave different answers. The results indicate that IEPlug helped them to learn more. For example, one participant stated: *'It explains why, but using external sources would be needed for more comprehensive explanation.'* It appears that this participant required more sources or hints about anti-phishing, perhaps similar to those offered by Google Safe Browsing.

The responses to the final statement indicate that both applications were not considered sufficiently reliable. Participants' open comments were mainly about i) the frequency of updating the blacklist, ii) the ability to make a successful detection, and iii) the whitelist maintenance of IEPlug.

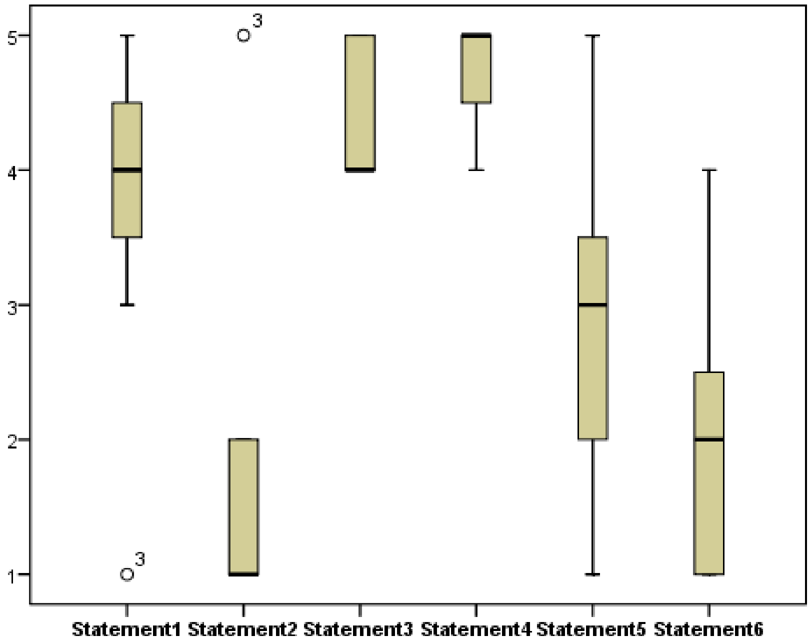


Figure 9. Questionnaire results from the Google Safe Browsing group.

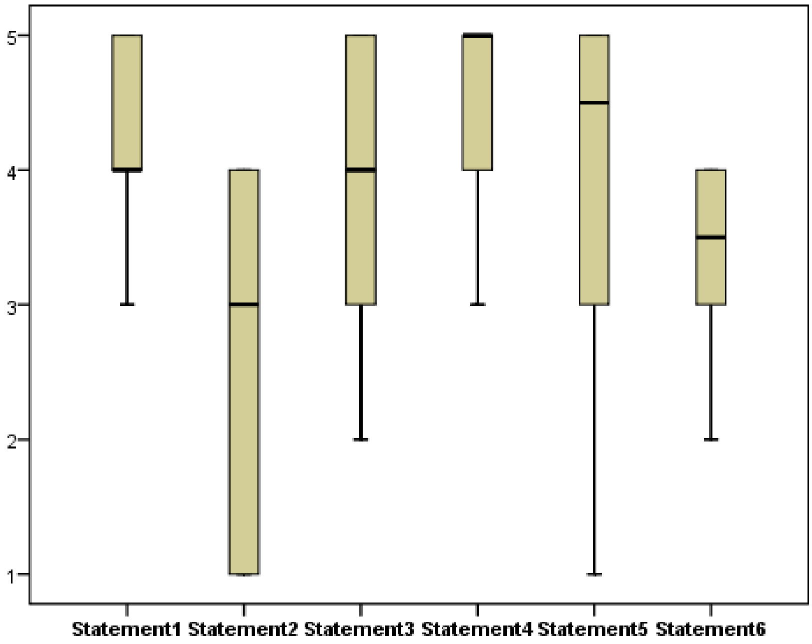


Figure 10. Questionnaire results from the IEPlug group.

5. Conclusions and future work

As online threats are becoming increasingly sophisticated, to protect ourselves against phishing, we need to adopt a contingency approach (Avison 1990). A contingency approach would mean applying a research and development methodology in circumstances where there are uncertain and unanticipated factors. This approach uses various techniques and tools (see, for example, Multiview by Episkopou and Wood-Harper 1985) to investigate these unpredictable factors from a variety of perspectives. In our case, we investigated more than one anti-phishing tool. Following the principles of a contingency approach, we selected two types of blacklist- and whitelist-based anti-phishing toolbars and designed our usability test to determine which toolbar type and which detection mechanism helps users to identify more phishing web pages. This was our initial intention, through testing users who understood how the toolbars operate.

However, we found no significant differences between the toolbars' detection accuracy. Furthermore, we discovered unexpected results concerning the learning effects of the tutorial sessions given to the users during the experiment.

During the test, all participants knew how to use the relevant application. Prior to the test, they were requested to read the associated tutorial and, if possible, try out the software. This test design resulted in the critical finding that neither of the tutorials helped to adequately educate the participants on how to use the tested applications to identify phishing web pages. To strengthen the educational effects of the tutorials on the users, we used an interim interview to observe users' behaviour and to provide us with more information on their personal strategies. This technique may also be appropriate for future anti-phishing software design methods. However, typical users do not read instruction manuals voluntarily (Kelley *et al.* 2012).

The nature of the usability test and our limited resources did not permit a large number of participants. Notwithstanding this, previous usability research results and studies show that even fewer than twenty participants are sufficient for a usability test if the purpose is to identify usability problems (Bevan *et al.* 2003, Faulkner 2003, Lindgaard and Chattratichart 2007).

There were further noteworthy research findings from our usability test. First, both applications did not appear to be sufficiently reliable for phishing prevention. Because it lacked analysis capability, the blacklist-based application was rarely helpful, and so was realized when we had to verify the authenticity of the tested web pages. On the other hand, the whitelist-based application's analysis capability was useful to participants, particularly because some used it to identify the actual domain names from the given web addresses. Naturally, this works only when a user knows that it is crucial to check and verify the domain name.

Pop-ups should not contain too much information, because this is unappealing and the information may not even be read by some inexperienced users; on the other hand, pop-ups should be as informative as possible for more demanding and experienced users. For example, our whitelist-based application pops up SSL CAs of safe websites, but this can also disturb users' regular browsing activities. This is contradictory for designers who may be considering equipping anti-phishing applications with analysis capability. On the one hand, analysis capability requires detailed information, but on the other hand, detailed information will always to some degree disturb users' online activities. How to balance this contradictory security and usability requirement at the design level remains an open question.

In general, a successful phishing-prevention methodology requires some technical skills. The average user appears to be unaware of the most relevant and useful information, but does not know how to find such information. It is also essential to determine how users can be motivated to verify the authenticity of websites. In his study on web spam, social propaganda, and the evolution of search engine rankings, Metaxas (2010) states that 'it is the users' right and responsibility to decide what is acceptable for them. Their browser, their window to cyberworld, should enhance their ability to make this decision.' For this reason, a good choice of technology may be a system that offers context-related help according to the various problems a user may come across. Nonetheless, it should also be remembered that even this type of system would not fully protect users. One of the reasons is that users may forget to check necessary web page identities and, thus, they are still vulnerable. Therefore, it would be a good idea to embed security design considerations in the procedures of frequent online activities such as payment transactions; this design strategy is likely to affect users' mental models that influence their online actions and their cognitive needs, thus encouraging them to learn more about 'virtual' dangers.

Finally, to ensure user protection and security, it appears to be better to use a system that combines the strengths of both whitelist- and blacklist-based applications, rather than using either a whitelist- or a blacklist-based application alone. This is because both these application types have advantages that, when used in a combined manner, assist the other's weaknesses. Hence, they complement each other, and this is the notable benefit derived from the adoption of our contingency approach. This approach allows multiple views of phishing instances that are highly dynamic, unpredictable, and increasingly sophisticated.

In this case, however, one benefit of using a solely whitelist-based application would be lost, which is that maintaining a whitelist does not require constant updates from the developers. Nevertheless, the users themselves would be able to maintain the whitelist. Furthermore, a suitable solution would be to use a whitelist-based application together with a blacklist-based application such as those built into Internet Explorer, Mozilla, and Opera browsers.

Our future research will involve the identification of significant security and usability features for defining adequate software quality criteria for the design of anti-phishing technology applications. Among other issues, the design of effective tutorials as well as formal and non-formal user training are major issues for consideration. In that respect, we fully agree with other researchers in the field who state that user education is fundamental: without it, the public largely trusts whatever they see and find on the web, regardless of its credibility. Furthermore, people should know *how* search engines work and *why*, and *how* information appears on the web. However, they should also have a trained browser that can help them to determine the validity and trustworthiness of information on the web pages they visit (Metaxas 2010).

Last but not least, we intend to conduct research comparing the effectiveness of using tutorials alone and tutorials in combination with interviews, to discover more about users' perceptions of usability and in relation to the identification of phishing sites.

Acknowledgements

The authors would like to thank all the participants of this experiment. Our thanks are extended to the researchers of the Tampere Unit for Computer–Human Interaction (TAUCHI) Research Centre of the University of Tampere, who were very supportive during this study.

REFERENCES

ANTI-PHISHING WORKING GROUP (APWG), 2011. *Phishing Attack Trends Report.* Available from: http://www.apwg.org/reports/apwg_report_h2_2010.pdf. [Accessed 16 December 2011].

Avison, D.E.,1990. *A contingency framework for information systems development.* Thesis (PhD). Aston University.

Bandura, A., 1977. Social learning theory, Prentice-Hall.

Bevan, N., Barnum, C., Cockton, G., Nielsen, J., Spool, J., and Wixon, D., 2003. The "magic number 5": is it enough for web testing?. *In: CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03).* ACM, New York, NY, USA, 698-699.

Blythe, M., Petrie, H., and Clark, J. A., 2011. F for fake: four studies on how we fall for phish. *In*: *the 2011 annual conference on Human factors in computing systems (CHI '11) ACM.* New York, NY, USA, 3469-3478.

Dhamija, R., Tygar, J.D., and Hearst, M., 2006. 2006. Why phishing works. *In: the SIGCHI Conference on Human Factors in Computing Systems (CHI '06),* ACM, New York, NY, USA, 581-590.

Dimensional Research, 2011. The Risk of Social Engineering on Information Security: A Survey of IT Professionals. Available from: http://www.checkpoint.com/press/downloads/social-engineering-survey.pdf. [Accessed 30 November 2012].

Egelman, S., Cranor, F. L., and Hong, J., 2008. You've Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings. *In*: *the twenty-sixth annual SIGCHI conference on Human factors in computing systems.* Florence, Italy, April 2008, 1065-1074.

Episkopou, D. M., Wood-Harper, A. T., 1985. The Multiview methodology:Applications and implications. *In: Bemelmans, T. M. A. (Ed.), BeyondProductivity:Information Systems Development for Organisational Effectiveness*. Amsterdam: North Holland.

Faulkner, L., 2003. Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods*, 35(3), 379-383.

Google Safe Browsing, 2007. Available from: http://www.google.com/tools/firefox/toolbar/FT3/intl/en. [Accessed in 16 April 2006].

Google Safe Browsing, 2007a. *Tutorials for Google Safe Browsing.* Available from: http://www.cs.uta.fi/%7Ell79452/Tutorials-GSB.doc [Accessed 16 April 2006].

Google Safe Browsing, 2007b. *Google Toolbar for Firefox Help Center*. Available from: http://www.google.com/support/firefox/bin/static.py?page=features.html&v=3. [Accessed 16 April 2006].

Hong, J., 2012. The State of Phishing Attacks. *Communications of The ACM*, 55(1), January, 2012. 74-81.

IEPlug, 2006. Available from: http://www.cs.uta.fi/~ll79452/ap.htm . [Accessed 16 April 2012].

Jakobsson, M., and Ratkiewicz, J., 2006. Designing Ethical Phishing Experiments: A study of (ROT13) rOnl auction query features. *In*: *the 15th annual World Wide Web Conference*. 513-522.

Kelley, P.G., Consolvo, S., Cranor, L.F., Jung, J., Sadeh, N., and Wetherall, D., 2012. A Conundrum of Permissions: Installing Applications on an Android Smartphone. *Financial Cryptography and Data Security Lecture Notes in Computer Science*, Volume 7398, 68-79.

Li, L. and Helenius, M., 2007. Usability Evaluation of Anti-phishing Toolbars. *Jounal of Computer Virology*, 2007(3), 163-184.

Li, L., Helenius, M., and Berki, E., 2011a. How and Why Phishing and Spam Messages Disturb Us? *In: IADIS International Conference ICT, Society and Human Beings 2011*, Rome, 239–244.

Li, L., Berki, E., and Helenius, M., 2011b. Evaluating the Design and the Reliability of Spam/Phishing Content Filtering Performance Experiments, *In: Software Quality Management 2011*, 339–357.

Li, L., Helenius, M., and Berki, E., 2012a. A Usability Test of Whitelist and Blacklist-based Anti-phishing Applications, *In: MindTrek Academic Conference 2012,* Oct 3–5, 2012, Tampere, Finland, 195–202.

Li, L., Berki, E., Helenius, M., and Savola, R., 2012b. New Usability Metrics for Authentication Mechanisms. *In: Twentieth International Conference on Software Quality Management*, Tampere, Finland.

Lin, E., Greenberg, S., Trotter, E., Ma, D., and Aycock, J., 2011. Does domain highlighting help people identify phishing sites? *In*: *the 2011 annual conference* *on Human factors in computing systems (CHI '11) ACM.* New York, NY, USA, 2075-2084.

Lindgaard, G., and Chattratichart, J., 2007. Usability testing: what have we overlooked? *In: the SIGCHI Conference on Human Factors in Computing Systems (CHI '07),* ACM, New York, NY, USA, 1415-1424.

Litan, A., 2004. Phishing attack victims likely targets for identity theft. FT-22-8873, *Gartner Research.*

Luca, A. D., Frauendienst, B., Maurer, M., Seifert, J., Hausen, D., Kammerer, N., and Hussmann, H., 2011. Does MoodyBoard make internet use more secure? : evaluating an ambient security visualization tool. *In*: *the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*, ACM, New York, NY, USA, 887-890.

Metaxas, P.T., 2010. Web Spam, Social Propaganda and the Evolution of Search Engine Rankings. *Web Information Systems and Technologies Lecture Notes in Business Information Processing*, 45, 170-182.

Microsoft, 2006. How to shop online more safely. Available from: http://www.microsoft.com/protect/yourself/finances/shopping_us.mspx. [Accessed 16 Febrary 2007].

Nielsen, J., 1993 . Usability Engineering, Morgan Kaufmann Publishing House.

Phish Tank, 2011. Available from: http://www.phishtank.com. [Accessed in 16 November 2011].

Phishing Protection, 2007. *Firefox phishing protection.* Available from: http://www.mozilla.com/en-US/firefox/phishing-protection/. [Accessed 16 April 2007].

Sheng, S., Holbrook, M., Kumaraguru, P., Cranor, L.F., and Downs, J., 2010. Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions. *In*: *the SIGCHI Conference on Human Factors in Computing Systems (CHI '10),* ACM, New York, NY, USA, 373-382.

Villamarín-Salomón, R. M., and Brustoloni, J. C., 2010. Using reinforcement to strengthen users' secure behaviors. *In*: *the SIGCHI Conference on Human Factors in Computing Systems (CHI '10),* ACM, New York, NY, USA, 363-372.

Wu, M., Miller, R. C., and Garfinkel, S. L., 2006a. Do Security Toolbars Actually Prevent Phishing Attacks? *In: the SIGCHI Conference on Human Factors in Computing Systems (CHI '06),* ACM, New York, NY, USA, 601-610.

Wu, M., Miller, R. C., and Little, G., 2006b. Web Wallet: Preventing Phishing Attacks by Revealing User Intentions. *In*: *the second symposium on Usable privacy and security (SOUPS '06).* ACM, New York, NY, USA, 102-113

Zhang, Y., Egelman, S., Cranor, L., and Hong, J., 2007. Phinding Phish: Evaluating Anti-Phishing Tools. *In*: *the 14th Annual Network and Distributed System Security Symposium (NDSS 2007)*. San Diego, CA, 2007, 79-92.

Study 6

Li L. (2012). Overview of User-centered Quality Assurance Methodologies for Anti-phishing Software and Phishing-resistant Systems, *Proceedings of Berki, E., Valtanen, J., Nykänen P., Ross, M., Staples, G., Systä K. (Eds), Quality Matters*, SQM 2012, Tampere, Finland, 20-23 August 2012, pp. 11-20.

Overview of User-centred Quality Assurance Methodologies for Anti-phishing Software and Phishing-resistant Systems

Linfeng Li

School of Information Sciences, University of Tampere,
Kanslerinrinne 1, 33014, Tampere, Finland
Linfeng.li@uta.fi

Abstract

In order to prevent the personal information from being stolen or misused, various anti-phishing software and phishing-resistant features are developed and deployed. However, the quality of this particular type of software varies, especially when phishing technologies are getting increasingly advanced. Due to the nature of phishing attacks, it is needed to guarantee the quality of anti-phishing software and phishing-resistant systems from end users' perspectives. In this position paper, we summarize four user-centred quality assurance methodologies to help improve the design of anti-phishing software and phishing-resistant systems. These methodologies were fully utilised during the research work that dealt with user requirement elicitation for anti-phishing software. In addition, these methodologies are strongly recommended when working for high-security and high-usability software projects.

1.0 Introduction

Online privacy, especially online identities, has attracted a lot of attention and research efforts for long time [1, 2, 3]. At the same time, attackers are constantly developing new techniques to steal online private information to gain profit, e.g. phishing attacks. Usually, the generic phishing scams spoof victims by sending persuasive and alluring emails with fraudulent content, and victims are asked to offer their online credentials to forged web services enclosed in phishing emails. Nowadays, sophisticated phishing scams use more advanced technologies. For example, the Trojan horse malware is used in the recent phishing scam so that to monitor the traffics between users' web browsers and the online service servers [4]. Sadly, these online threats cannot be effectively prevented, even though various security features and products were developed and deployed [5]. This implies that quality improvement of anti-phishing software was also imperative [6, 7].

To assure the software quality, software engineers are assigned to verify whether the developed software meets the requirements from its stakeholders, including managers, software designers, and end users. For security and anti-phishing applications, software designers should particularly concentrate on users' preferences [7], and find out *what end users expect from anti-phishing software and phishing-resistant systems*. To answer this question, we summarize user-centred quality assurance methodologies that we have been used to deal with the quality metrics research for anti-phishing software. In the following section, we present four quality assurance methodologies. Besides that, we also list the findings of each methodology, and analyze their advantages and disadvantages. After that, we conclude the research and the contributions of these methodologies, and foresee the research work on the quality assurance of anti-phishing software and phishing-resistant systems.

2.0 Quality Assurance Methodologies for Anti-phishing Software and Phishing-resistant Systems

In this section, we introduce four main user-centred quality assurance methodologies for anti-phishing and phishing-resistant systems, including misuse cases, end user survey, usability evaluation research, and user behaviour modelling. In the following part, we explain these methodologies and present our key findings in these studies. Besides, the advantages and disadvantages of each methodology are discussed.

2.1 Misuse case methodology

Misuse case methodology was firstly introduced by Sindre and Opdahl [8], which aims to test and evaluate the security breaches in the design and the use cases at requirement stage. In general, a misuse case is one inverse use case where the user requirements are defined. Since misuse cases are generated by finding out the vulnerabilities in use cases, it requires the use cases as the inputs. To investigate the potential security vulnerabilities in the collected use cases, Sindre and Opdahl concluded the misuse case specification in misuse case test methodology. Similar to use cases, some fields must be specified in misuse case specifications. The key fields include basic paths, alternative paths, capture points and extension points. Basic path is the primary method by which a crook is able to abuse a certain use case. Alternative paths list more possible methods to misuse the use case. Capture points describe how the misuse can be possibly stopped or detected. Extension points present other possible misuse cases that may be taken advantage of by crooks.

Besides these key fields, there are also some fields to present the roles, the tracing information, the rationales and threat consequences of the misuse cases, e.g. trigger, preconditions, related business rules etc. In addition, descriptive fields are included in misuse cases specifications, e.g. misuse case scope, profile, level, stakeholders etc. The steps of design misuse case methodology are as below:

1. Design the use cases of the system;
2. impersonate a misuser, who intends to compromise the system;
3. design the misuse for a specific use case;
4. find a countermeasure for a misuse case;
5. judge whether the countermeasure is vulnerable; if yes, go to step 3, otherwise go to the next step;
6. find whether there is another possible vulnerability or misuse; if yes, go to step 3, otherwise security requirements elicitation ends.

With misuse case methodology, requirement engineers are able to elicit security requirements. In our paper [9], we found that the reported phishing scams can be adapted as misuse cases to elicit security requirements. In addition, these misuse cases can be reused for future development or software development and design. However, its limitations are also obvious. First of all, phishing attacks are evolving to be more advanced. The future phishing attacks cannot be effectively and efficiently predicted or involved with this method. This is because the origin of successful and evolving phishing attack cannot be presented with misuse cases. Secondly, managing and organizing these elicited misuse cases is costly. In the evolving environment of phishing and its preventions, any new phishing techniques or new security improvement may result in dramatic changes on misuse case specifications. To update and trace these changes, it may take a great amount of maintenance efforts and resources from software development teams.

2.2 End User Survey

Surveys are widely used to collect end users' feedback towards the investigated objects. This methodology is also useful for collecting users' requirements on the experiences of phishing and its preventions. In general, surveys are designed to ask a series of questions to find out users' opinions on specific topics in a structured or non-structured way. For the research on phishing and its preventions, we have to collect more data from end users. This is because firstly, survey participants may come from different countries and have their unique experiences on phishing attempts, so collecting phishing data from real-world can investigate multilingual issues in anti-phishing research. In addition, security software should help users solve the most important problems that phishing brings. By analyzing the collected phishing emails from users, researchers are able to know what types of phishing attempts are the most annoying ones. Moreover, with phishing emails from survey participants, we are able to understand users' feelings and emotions more precisely. As a key feature of human-centred quality design, users' feelings and emotions reflected when receiving phishing/spam emails should be investigated.

Different individuals may feel differently when receiving the same type of phishing emails. By taking phishing/spam samples, we are able to observe and pinpoint how the survey participants are disturbed by phishing emails.

In our research [10], we investigated a number of phishing/spam samples from our invited participants who were from different countries and extracted useful characteristics of these emails in order to classify them. We also designed and implemented a survey to disclose how these invited participants feel about the disturbance of phishing/spam emails. From the investigation on the reported phishing/spam samples, we discovered that phishing/spam emails were not always very distinct from legitimate ones. In addition, friends' email addresses could easily convince victims to believe the authenticity of the emails. The language barrier also prevented successful phishing. Moreover, due to the lack of an opt-out option, participants classified the news from subscription services as phishing/spam emails.

From the survey, we collected other kinds of valuable feedback. For example, some participants replied that spam messages were time-consuming and annoying for daily life. In addition, we found that different people may have different interests so that to have different criteria to define spam/phishing emails. Furthermore, the participants gave suggestions to use separate email address from their working emails.

With the analysis of investigation results and survey feedback, we believed that a drastic and influential approach towards email users' protection needs to emerge three important issues: (i) software user psychology; (ii) human-centred software design quality criteria, and (iii) software/email exploiters' cognitive profiles, even though the number of email samples and survey participants in this experiment is limited.

Even though these findings are valuable and user surveys are helpful for phishing-related research, some of its disadvantages are revealed. These subjects invited in the survey are not always experts, and they are not always able to correctly describe the process and the methods of identifying phishing attempts. As a result, a variety of responses from the great amount of survey participants on phishing are still difficult to manage. Therefore, structured surveys are preferred.

2.3 Usability Research Methodologies

Usability, as a property in software quality, plays an important role in software quality assurance. Usability research methodologies, including heuristic evaluation and usability test by end users, are to analyse and evaluate the usability problems and flaws in the existing products so that to figure out how to improve these products or enhance user experiences. In our heuristic evaluation research for anti-phishing [11], we first revised the system heuristics [12] to fit into the evaluations on anti-phishing toolbars. Afterwards, we invited usability experts to join our heuristic evaluation so as to make the tested software and heuristics evaluation materials familiar. During the evaluation, we collected the comments and findings by the invited experts. After the evaluation, we grouped them and presented the valuable evaluation results as follows:

1. Main user interface of the toolbar.

- The status of the toolbar should be shown appropriately.
- The application interface should be simple enough so that it is easy to understand and it does not take too much space from the browser's interface.
- Frequently used and important functionalities, such as configuration settings and viewing the website identity analysis result, reporting a suspicious or misjudged web page, should be convenient enough to be found.

2. Warnings.

- A warning should be able to stop users' faulty visits properly.
- The undetermined page requires to be notified to users.
- Innocent pages should be indicated respectively.
- A double warning should be used in case an erroneous choice is made. If a user accidentally selects a choice that leads to visiting a phishing website, a second warning should be available to correct the mistake.

3. Help system.

- Client side anti-phishing application should be able to help users in any phishing prevention occasions.
- The ways of showing help for different anti-phishing occasions may not be the same.
- Too much text information in the help system may confuse users.

These findings are helpful for the designers to improve anti-phishing software, but its limitations are obvious, e.g. the lack of end users' feedback, the limited number of usability experts and the limited number of tested software designs.

Besides this heuristic evaluation, a usability test was conducted with real end users involved to compare different phishing web page detections. Multiple phishing web page detections are available, and usually they are either blacklist based or whitelist based software. To compare these two main detection mechanisms from usability perspective, we organized a usability test with 20 participants involved [13]. The usability test was conducted in a controlled environment, and followed the procedure as in Figure 1, which consisted of four parts. The first part was to present the participants with the tutorial information on how the tested anti-phishing toolbar works. Then, these participants were asked to identify phishing pages from a list of web pages consisting of detected phishing pages, undetected phishing pages and authentic web pages. After that, the participants were interviewed and were given the correct knowledge on how to use the tested anti-phishing toolbars. After the interview, the participants checked the authenticity of another set of prepared web pages during the second test session.

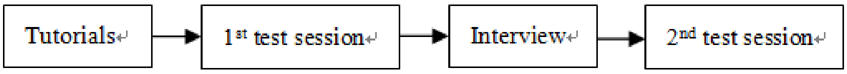


Figure 1. The usability test procedure

In the usability test, we found that correctly learning how to use the test anti-phishing software significantly impact users' judgements of identifying phishing pages, even though no significant difference was found between the blacklist-based phishing preventions and the whitelist-based one. Besides that, we also observed that: 1) most of our participants (17 out of 20) cannot correctly use the unique properties of web pages, including domain names, privacy policies, secure connection pad lock icons etc. 2) Some of our participants (3 out of 20) were not aware of the phishing attacks and online spoof threats which negatively impact the user experiences of tested anti-phishing software.

With usability research methodologies, we efficiently figured out the usability flaws in the tested phishing preventions. Particularly, the usability tests with real end users involved are even more straightforward and more effective to discover how users interact with these anti-phishing applications. For the designers and software developers of phishing preventions, these findings are valuable references. However, to guarantee the quality of the anti-phishing software, these references are not adequate. This is because 1) limited types of phishing scams were involved in these usability research experiments. Even though we tested phishing web pages with similar domain names and page layout, the sophisticated phishing attempts are not taken into use in the experiments. Therefore, it is still unknown how to guarantee the quality of anti-phishing software towards evolving phishing scams. 2) Many user behaviours were observed and collected during the tests. E.g. some participants preferred to check domain names to identify authentic web services, but some of our participants use web contents as reliable identities of web pages. In this case, how to present reliable and usable web page identities still require further investigation. These diverse user behaviour patterns also indicate the need to conclude an abstracted and concise model to describe the quality criteria of anti-phishing software. To generate software quality criteria, we build a user behaviour model to elaborate the key states and the whole process of how users make decisions to identify phishing scams.

2.4 User Behaviour Modelling Methodology

Various meta-modelling methodologies are applied in software development, e.g. CDIF[14], GOPRR[15]. These methods define the objects, relationships, and their properties. However, these meta-modelling methodologies are not applicable or cost-effective when we study the complicated and dynamic user behaviours in the constantly evolving phishing context. To model user behaviours in the phishing context, we used finite state machine (FSM) theory [16]. To facilitate the modelling and improve the quality of the model, a concurrency verification tool was applied, Labelled Transition System Analyser [17]. With this tool, we defined the user behaviours in the phishing context in Appendix A. In our model, we easily understand that users make wrong decisions and spoofed by phishing scams when they lack correct knowledge on the information phishers try to convey. In addition, it is highlighted that users can hardly be convinced by phishing scams if their security knowledge is reinforced, and are consequently the risks of being spoofed are eliminated.

Different to other meta-modelling methodologies, FSM emphasizes and describes process of state transitions. This is a clear advantage for modelling user behaviour patterns in the phishing context. Firstly, user behaviour patterns may vary in the evolving environment of phishing attacks. For example, some users may get infected by malware when they are visiting malicious websites where malware is hosted, and some of the victims may mistakenly run Trojan horse programs when they are sharing pictures where malware is added. In both cases, the installed malware is able to monitor the victims' online activities and steal their credentials via different channels. Meta-modelling methodologies other than FSM cannot present a concise and abstract picture to understand how users are spoofed to install these multiple channels. Instead, these meta-modelling methodologies will probably result in detailed but very complicated models of user behaviours. Consequently, these models are not general or valuable enough to come up with usable preventions against future phishing attacks.

In addition, FSM designed by LTSA tools are able to be verified easily. The logics in FSM can be run by LTSA in a linguistically expressive and readable way. In order to model user behaviours, we need not only a highly abstracted modelling methodology, but also an expressive model to depict the transitions among the defined states. The transitions are highlighted and undefined transition deviations can be easily identified when running automatically with LTSA. To apply FSM to the practical phishing preventions, some limitations of FSM should be mitigated. Our FSM modelling is abstracted and focusing on the vulnerabilities of user behaviours. To design usable and secure phishing preventions, the legacy systems must be taken into consideration. For example, to reinforce the usability and security quality of existing online-banking systems, the behaviours of these systems should be also abstracted and integrated with our FSM. For the purposes like this, an automatic tool to abstract the legacy systems is needed so that to supplement the limitations of our FSM.

3.0 Conclusions and Future Work

Phishing takes advantages of security and usability vulnerabilities of online services with technology and social engineering methods. Technology enables phishing attacks spread over the Internet; and social engineering attacks help convince victims with fraudulent information. Although various anti-phishing applications are deployed, their quality is not always satisfactory [18]. In this position paper, we review four user-centred quality assurance methodologies and their findings that provide guidelines to guarantee the quality of anti-phishing and phishing-resistant systems. In addition, these methodologies are proved to be valuable to collect these user requirements and to be useful to abstract the diversity of user behaviours to discover how users make decisions in the phishing context, and how to help potential victims to prevent phishing scams.

In our study, we emphasize the importance of anti-phishing research from software engineering perspective. Besides collecting users' preferences and requirements on phishing preventions, these user-centred quality assurance methodologies are able to be reused by other security- and usability-related software projects.

In spite of the clear advantages of these user-centred methodologies, it is not ignorable that future research efforts are needed to upgrade these methodologies. For example, a software management tool is needed to effectively and efficiently organize and maintain the growing number of misuse cases; how to integrate the legacy system models with extracted user models requires future research efforts. All in all, the essence of user-centred quality assurance methodologies is to elicit users' requirements on phishing preventions. With these collected requirements, reliable test cases can be efficiently generated for future anti-phishing software. With the help of these methodologies, researchers are also able to deeply understand the root cause of phishing threat and to further protect end users with high quality anti-phishing software and phishing-resistant systems.

4.0 References

1 Berki E., & Jäkälä M., Cyber-Identities and Social Life in Cyberspace. Hatzipanagos, S. & Warburton, S. (Eds) Social Software and Developing Community Ontologies (London: Information Science Reference, an imprint of IGI Global). pp28-40. London, 2009

2 Dhillon G. & Moores T., (2001). Internet privacy: Interpreting key issues. Information Resources Management Journal, 14, 4, 33-37.

3 Warren M., & Hutchinson W. (2002). Cyberspace Ethics and Information Warefare, Social Responsibility in the Information Age: Issues and Controversies. Idea Group Publishing 2001, ISBN-10: 1930708114

4 Wyke J., (2012). What is Zeus, technical paper, http://www.sophos.com/en-us/why-sophos/our-people/technical-papers/what-is-zeus.aspx (visited January 2012)

5 Hong J., (2012). The State of Phishing Attacks. Communications of The ACM, 55(1), January, 2012. 74-81

6 Berki E, Isomäki H., Salminen A., Quality and Trust Relationships in Software Development, proceedings of the Software Quality Management (SQM) International conference. pp. 381-388. Staffordshire, Tampere. 2007

7 Berki E., Georgiadou E., Holcombe M., (2004). Requirements Engineering and Process Modelling in Software Quality Management— Towards a Generic Process Metamodel, Software Quality Journal, 12, 3, 265-283

8 Sindre G. & Opdahl A., Templates for Misuse Case Description, 2001, http://www.ifi.uib.no/conf/refsq2001/papers/p25.pdf. (visited November 2006)

9 Li L., Helenius M., Berki E., Phishing-Resistant Information Systems: Security Handling with Misuse Cases Design, proceedings of Software Quality in the Knowledge Society, pp389-404. Tampere 2007

10 Li L., Helenius M., Berki E., How and Why Phishing and Spam Messages Disturb Us? proceedings of IADIS International Conference ICT, Society and Human Beings 2011. pp239-244, Rome, 2011

11 Li L. & Helenius M., (2007). Usability Evaluation of Anti-phishing Toolbars, Journal of Computer Virology, 2007, 3, 163-184

12 Pierotti D., Usability Techniques: Heuristic Evaluation - A System Checklistm, http://www.stcsig.org/usability/topics/articles/he-checklist.html, (visited October 2006)

13 Li L., Helenius M., Berki E., (2012). A usability test of whitelist and blacklist-based anti-phishing applications, MindTrek Academic Conference.

14 Berki E., 2001. Establishing a Scientific Discipline for Capturing the Entropy of Systems Process Models. CDM-FILTERS. A Computational and Dynamic Metamodel as a Flexible and Integrated Language for Testing, Expression and Re-engineering of Systems. Ph.D. Thesis. Faculty of Science, Computing and Engineering. University of North London, 2001.

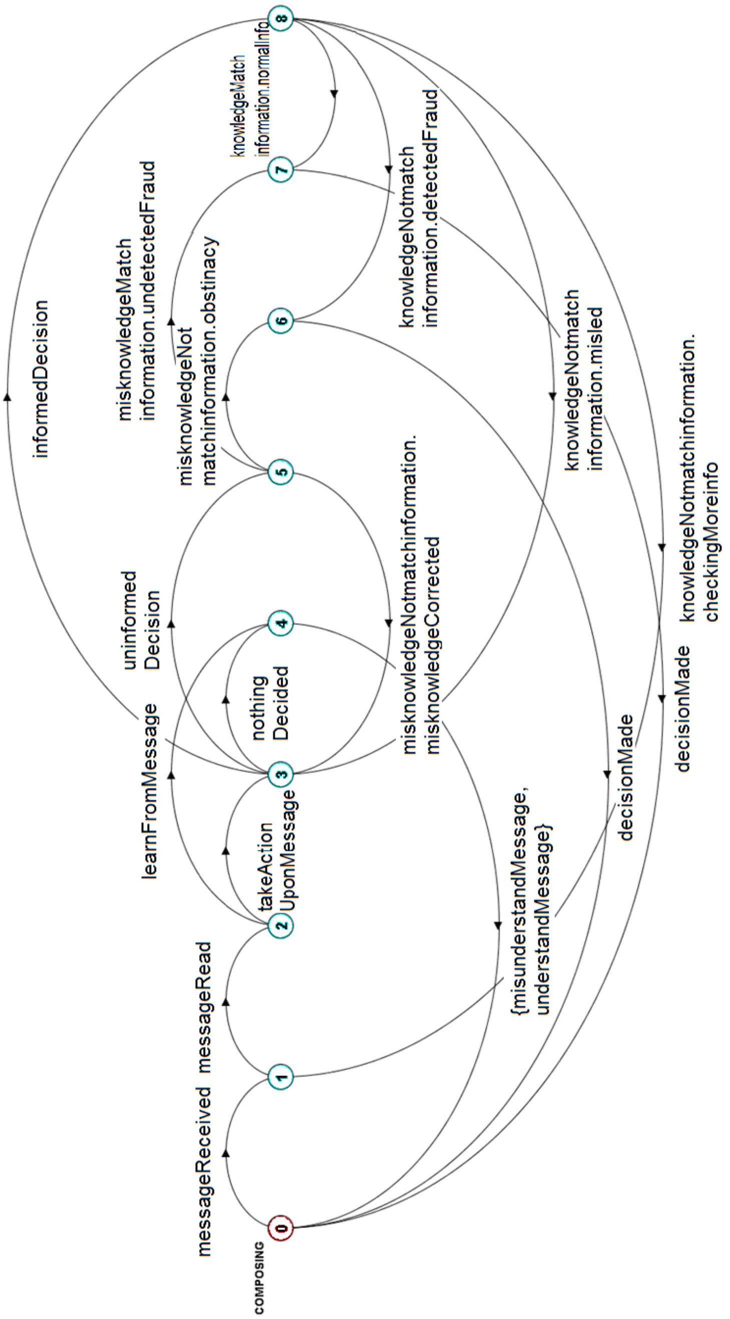
15 Kelly S., GOPRR Description, http://metaphor.it.jyu.fi/a1goprr.html, (visited Oct., 2011)

16 Sipser M., (1997). Introduction to the Theory of Computation, PWS Publishing Company 1997, ISBN-10: 053494728X

17 LTSA, Labelled Transition System Analyser, http://www.doc.ic.ac.uk/ltsa/, (visited November 2011)

18 Zhang Y., Egelman S., Cranor L., and Hong J.. Phinding Phish: Evaluating Anti-Phishing Tools. Carnegie Mellon University, CyLab Technical Report. CMU-CyLab-06-018, 2006, http://www.cylab.cmu.edu/default.aspx?id=2255, (visited November 2006)

APPENDIX A



COMPOSING
0
1
2
3
4
5
6
7
8
messageReceived
messageRead
takeAction UponMessage
learnFromMessage
nothing Decided
uninformed Decision
informedDecision
misknowledgeNot matchinformation.obstinacy
misknowledgeMatch information.undetectedFraud
knowledgeMatch information.normalInfo
knowledgeNotmatch information.detectedFraud
knowledgeNotmatch information.misled
misknowledgeNotmatchinformation. misknowledgeCorrected
{misunderstandMessage, understandMessage}
decisionMade
decisionMade
knowledgeNotmatchinformation. checkingMoreinfo

Figure 2. FSM diagram generated by the LTSA script of user behaviour in phishing context

Study 7

Li L., Berki E., Helenius M., Savola R. (2012). New Usability Metrics for Authentication Mechanisms, *Proceedings of Berki, E., Valtanen, J., Nykänen P., Ross, M., Staples, G., Systä K. (Eds), Quality Matters*, SQM 2012,, Tampere, Finland, 20-23 August 2012, pp. 239-250.

New Usability Metrics for Authentication Mechanisms

Linfeng Li[1], Eleni Berki[1], Marko Helenius[2], Reijo Savola[3]

[1]School of Information Sciences, University of Tampere,
Kanslerinrinne 1, 33014, Tampere, Finland
Linfeng.li@uta.fi, Eleni.Berki@uta.fi
,

[2]Department of Communications Engineering, Tampere University of Technology,
Korkeakoulunkatu 1, 33720 Tampere, Finland
marko.t.helenius@tut.fi

[3]VTT Technical Research Centre of Finland,
Oulu, Finland
Reijo.Savola@vtt.fi

Abstract

Authentication mechanisms face various online threats, such as malware programs and phishing scams, which challenge the existing authentication mechanisms' security and usability. Authentication mechanisms with poor usability may easily be exploited by attackers and cause severe security risks. In order to guarantee the design quality of authentication mechanisms, we propose a set of usability metrics. With the help of the proposed usability metrics, software developers and security and usability experts will be able to evaluate and measure usability and security quality of authentication mechanisms.

1.0 Introduction

Many convenient services are available on the Internet. The users of these services often need to provide private information, like frequent web browsing details and personal data. In this way, they expect that the quality of online services can be improved [1, 2, 3], as it is promised by the service providers. When it comes to e-commerce, even more sensitive information is required, e.g. credit card numbers and bank account information. To protect personal online data and services, user authentication mechanisms are applied.

Traditionally, users are authenticated by a user name - password pair to sign into online services. One would expect that these authentication mechanisms are secure enough. However, frequently reported phishing scams show that these traditional authentication mechanisms are not particularly good at preventing online fraud. First and foremost, this happens because phishing attacks get increasingly sophisticated all the time. For example, an increasingly sophisticated Trojan horse malware has recently been employed to steal personal online credentials, i.e. login user names and passwords [4]. Secondly, more and more smart devices and services are connected to each other on the Internet to build the so called Internet of Things (IoT). This evolutionary need requires from the users to possess correct knowledge on how to correctly operate, manage and configure these devices and networks.

In order to prevent users from being spoofed in such a complicated phenomenon at the very beginning of user experiences on online services, it is required to devise a usable authentication mechanism. When developing a usable authentication mechanism, there is no need to carry out a completely new design. Many research efforts have proved that improving and maintaining the existing authentication mechanisms is also valuable [5, 6, 7, 8, 9], and often the most cost-effective solution. These studies tend to put more efforts on security and less on usability.

In this paper, we propose a set of high-level usability metrics as guidelines for the designers of online authentication mechanisms based on the analysis of the existing authentication mechanisms and online identity theft cases. The viewpoint is high level for two reasons: Firstly, the definition of usability is too general; more detailed definitions are obviously needed to gauge enough user experience [10]. In addition, different authentication mechanisms have their special designs, like input methods, interactions between users and online services, and the list goes on. Too detailed metrics may also result in non-comparable findings.

In the following, we preliminarily analyse the characteristics and properties of a representative set of authentication mechanisms. After that, we analyse the different properties of these authentication mechanisms from the usability perspective. Based on this analysis and the existing anti-phishing research findings, we deduce a set of usability metrics based on the analysis and evaluate the advantages and disadvantages in these existing authentication mechanisms.

2.0 Analysis of Existing Authentication Mechanisms

In security metrics, the main properties contributing to authentication effectiveness, or strength, are (i) the identity effectiveness and (ii) the authentication mechanism effectiveness [11]. Uniqueness of identity contributes essentially to the former category, and in the latter one, reliability and integrity of the mechanism is crucial. In the following, we discuss how usability of identity token and authentication mechanism also contribute to the end user authentication strength.

For the purposes of analysis, we selected several representatives of services with critical authentication solutions based on *popularity*, *security* (or *privacy*) *sensitivity*, and *use frequency*. This selection was narrowed down into *email, online banking, online payment, instant messengers* and *door entrance control* systems. Based on the big number of online active users and different personal information collected for authentication mechanisms, we selected the following services: 163 email service [12], Nordea bank [13], Alipay [14], QQ messenger [15], and University of Tampere entrance systems [16].

In general, widely known authentication factors are:

- "What you know" means that a user knows something that is used for authentication.
- "What you have" means that a user must possess something that is used for authentication.
- "What you are" means that a user's personal characteristics are used for authentication.

The authentication becomes stronger if two or three factors are used in the authentication process instead of one, due to the increase of identity uniqueness by each factor. To facilitate our analysis, we use these three factors to illustrate the characteristics of authentication mechanisms.

In the following, we discuss additional significant properties for authentication mechanisms. Each of these properties must play (i) an important and (ii) independent role to the effectiveness and/or functionality of authentication mechanisms. For example, when a user registers to an online recruitment service, he/she needs to provide user name, password, contact information, and some other details like, CV. We classify these data into two independent properties: (i) login information, i.e. user name and password, and (ii) registration information, including all of the personal information that is required for registration. Both groups of data are critical for authentication mechanisms, and do not directly affect another one. We list the identified properties, with their main characteristics from the user perspective in Table 1.

Table 1. The factors and their characteristics of authentication mechanisms from the user perspective

Factor: What you know	
Properties	***Characteristics from the User Perspective***
User name - password pair	1. Easy to remember for one service 2. Difficult to remember different sets of User name - password pair for different services 3. Sensitive. Some people, especially those who are aware of the importance of online identity security, may get easily alerted when user name and password are asked.
Registration	1. Confidential. The information is very personal.

Information	2. Personal. 3. Authentication candidate method. It is possible to use it in authentication process, e.g. security question, email address.
Login status	1. Straightforward. Information is given when a user is logged in. 2. Replicable. People can replicate the login status. 3. Passive. End users can not check their service login status.
Security status	1. Technical information involved. Sometimes, the security information is presented in very technical way. 2. Implicit. Sometimes, the security issues are not highlighted during registration and authentication for the sake of better user experience.
Look and Feel	1. Replicable. People can replicate the look and feel. 2. Straightforward. Information is given when needed. 3. Visual difference from others. Different companies and services intend to use different look and feel.
Factor: What you have	
Properties	***Characteristics from the User Perspective***
Smart mobile phone	1. Computation power. Some applications can run on the smart phone. 2. Multiple communication channels and connections. Phone calls, SMS, WIFI, 3G. 3. Compact. Easy to carry and use.
E-token (e.g. RFID, Radio Frequency Identification, SIM card)	1. Compact. Easy to carry and use. 2. Unique. Different services need different tokens 3. Technical. Some services may require some maintenance and management.
Paper token	1. Compact. Easy to carry, deliver and use. 2. Unique. Different services need different tokens. 3. Replicable. Easy to copy.
USB key	1. Compact. Easy to carry 2. Technical. People may need to learn its functioning and malfunctioning.
Factor: What you are	
Properties	***Characteristics from User Perspective***
Biometrics (e.g., fingerprint, voice, iris)	1. Individually highly unique. Different users have different characteristics. 2. Recordable. 3. Technical limitations. Cannot be used reliably for all users, e.g. finger prints cannot be always read, and voice may vary. There is a risk of false negatives and false positives
Users	1. Knowledgable. Users have some knowledge, but not always adequate.

	2. Educatable. Users can experience and learn new things by themselves. 3. Making mistakes. Users may make mistakes when they are learning by themselves.
Memberships	1. Sociable. People have relationships in family and society. 2. Communicative. People need to get contact with each other. 3. Preference. People have preferences in different area.

This preliminary analysis illustrates that each property of the collected authentication mechanisms has their specific characteristics. In general, these characteristics are not direct usability metrics, but we can use them to define essential properties contributing to actual usability metrics.

First of all, to be authenticated, users must provide registered secrets to verify themselves. These secrets could be defined by users, possessed by them, or be generated automatically by certain software or hardware logics, depending on the authentication factor(s) utilized. From the usability perspective, we need to measure how easily and effectively these secrets can be remembered, used, reused and maintained with sufficient confidentiality. Easiness is concerned with efficiency that means users should not spend too much time or effort on these secrets and users' identities. Effectiveness means that users are able to correctly use these secrets and their identities. According to our analysis, the improvement is usually applied by phishing-resistant system features, e.g. login from unusual regions [15], bank transaction notifications [14] etc. Furthermore, users have to learn to use some security devices. The effort of correctly learning to use new devices as a part of the authentication mechanisms should be taken into consideration.

Besides of these properties in authentication mechanisms, online threats exploiting the usability vulnerabilities in authentication should not be ignored. The common attacks of online authentication mechanisms are phishing web pages and man-in-the-middle attacks [5]. The phishing pages mimic the visual and written contents of authentic web services, including look and feel, domain names and URLs. From security and usability perspectives, only the visual effects cannot provide a reliable and usable user interface to identify the authenticity of web services. Man-in-the-middle attacks try to interrupt the online communications so that to collect credential information. Usability metrics for authentication mechanisms should be used to mitigate these problems, e.g. non-replicable authentication user interface and non-intrusive authentication services. An important issue in preventing phishing is also that there is a reliable and usable help system [17]. Phishers exploit users' knowledge gaps in order to convince them to use a falsified service. A reliable and usable help system should be able to educate users and provide positive user experiences especially in a complicated network environment [18], like IoT network. According to the research findings from [19], we realized that users prefer to find the information from web pages to identify the authenticity of online services. Therefore, it is also needed to effectively provide learnable and reliable design of authentication mechanism to help users verify their authenticity.

3.0 Proposed Usability Metrics for Authentication Mechanisms

In the previous section we were not able to compare which authentication mechanism is better from usability perspective. We propose a set of usability metrics for authentication mechanisms to facilitate this. Based on the analysis of factors in authentication mechanisms, we suggest the following usability metrics:

1. Customizability
2. Learnability
3. Credential information maintenance for multiple online services
4. Efficiency
5. Quality of help system
6. Replicability
7. Effectiveness
8. Non-intrusiveness
9. Cost-effectiveness

Customizability refers to that users are able to define their own credential information for authentication. For example, users can often use their own defined user names and passwords as credential information to sign in an email service. With this customizable credential information, it is easy for users to remember them. In the past research, customizable credentials raised many arguments on security and personal information used in credentials. Security experts believed that binding credentials to personal information is a bad idea [20], but usability researchers suggest improving password selection mechanisms to be memorable and secure [21]. In our usability metrics, we use customizability to describe how easy credentials can be remembered, and leave the secure credential design issues to the usability metric, *Quality of help system.*

Learnability means that users are able to easily learn how to use the authentication mechanisms. For some authentication mechanisms, there are many devices involved, e.g. paper-based tokens, USB keys. Even though they are easy to use, it takes time to learn how to *correctly* use them. More factors in authentication mechanisms may enhance the security, but regarding usability, the learnability aspect should be taken into account.

Credential information maintenance for multiple online services concerns the amount of efforts that end users need to maintain the credential information for multiple online services, e.g. emails, online banking, online payment and social networking. Every online service has its own rule to define the passwords. In some services, these rules are strictly enforced, whereas in some not. In this case, to define and manage these passwords may take some amount of efforts. Moreover, users may reuse the same credentials for different services, and credential information reusability may increase the vulnerability and result in high maintenance cost of victims. To compare the maintenance efforts, we define this usability metric.

Efficiency exams how many time and labour efforts are used when users are to be authenticated. For example, users sometimes need to type in their defined password or random numbers generated by smart devices. In both cases, it is possible to type in erroneous characters. These typos may give negative user experiences. To evaluate the usability in this perspective, we use efficiency to describe it.

Quality of help system refers to how easily users can be assisted when using authentication methods. For example, users may forget passwords or user names, or users may need to learn about authentication methods, or users may do not know how to design a secure credential. In these cases, a convenient help system can have positive effects on user experiences of authentication mechanisms.

Replicability metric describes whether the information involved in authentication mechanisms can be replicated. The most popular phishing scam is to copy the look-and-feel of web authentication mechanisms. If some of the information is not replicable, some phishing scams can be prevented.

Effectiveness calibrates how effectively authentication mechanisms and users can identify each other. Traditional password and user name pairs can only identify users. To prevent abuse of users' identities, users should be able to effectively identify the authentic online services and authentication mechanisms. In our study, phishing-resistant system features are good examples to increase the effectiveness of authentication mechanisms.

Non-intrusiveness verifies the integrity of the whole authentication process. Phishers employ man-in-the-middle attacks [5] to intervene in authentication mechanisms. If an authentication mechanism can prove its integrity, man-in-the-middle risks can be eliminated.

Cost-effectiveness evaluates whether the extra cost of authentication mechanisms can lower the probabilities of financial losses and bring more pleasant user experiences. Indeed, more security enhancements bring more reliable authentication mechanisms, but the cost of these extra services and devices are high at the same time. Do users feel worth of paying more service fee to get this enhanced security? Do users feel the risks can be effectively mitigated after paying the extra security service fee? All of these questions can be involved in the surveys to collect users' feedbacks. If the credentials are hacked and people abandon the service, the cost of “shame” to the service provider is significant. However, the analysis of this cost is out of the usability study scope of authentication mechanisms.

4.0 Evaluation of Authentication Methods with the Proposed Usability Metrics

To verify that our proposed usability metrics are valuable and applicable for various technical authentication methods, we evaluated them within the selected online authentication mechanisms. Due to the fact that usually the authentication methods play the key roles in the authentication mechanisms, we only gauged the usability of these methods and resulted in comparable findings as in Table 2.

There are five authentication methods. The first one is the traditional password and user name combination while the second one is the traditional one-time pad on a paper meaning a list of one-time passwords used by many online banks. The third one is e-token with computing power, meaning an electronic device to generate one-time passwords or secrets for authentication. This includes USB keys. E-token with no computing power means RFID tag or a piece of electronic circuit, where the identification information is stored.

The fifth one combines OpenID and GBA (Generic Bootstrapping Architecture) [22]. In GBA an application, e.g. in a smart phone, establishes a shared secret between the application and the phone network operator while OpenID is a commonly used single sign-on system. The idea of the combination is that a user does not need a user name - password pair, since the network operator acts as a trust anchor, and a shared secret on the SIM card is used to automatically establish a shared secret for single sign-on.

Table 2. Comparing different authentication methods with proposed usability metrics

Authentication Methods	***Usability Evaluation***	
	Pros	***Cons***

password - user name pair	Customizable;
	Easy to learn;
	Convenient help system;
	Low cost
		Not easy to maintain for multiple services;
		Not efficient;
		Easy to be replicated;
		Not effective to help users identify online frauds;
		Vulnerable to man-in-the-middle attacks;

one-time pad on paper	Easy to learn;
	Efficient;
	Low cost
		Not customizable;
		Easy to be replicated;
		Costly help system;
		Not easy to maintain for multiple services;
		Not effective to help users identify online frauds;
		Vulnerable to man-in-the-middle attacks

e-tokens
with computing power
	Easy to learn;
	Efficient;
	Not easy to be replicated;
	Moderate cost
		Not customizable ;
		Costly help system;
		Not effective to help users identify online frauds;
		Not easy to maintain for multiple services;

		Vulnerable to man-in-the-middle attacks;

RFID
with no computing power
	Customizable;
	Easy to learn;
	Efficient;
	Not easy to be replicated;
	Not vulnerable to man-in-the-middle attacks;
	Low cost;
		Not effective to help users identify online frauds;
		Not easy to maintain for multiple services;
		Costly help system;

OpenID and GBA	Customizable;
	Easy to maintain for multiple services;
	Efficient;
	Not vulnerable to man-in-the-middle attacks;
	Low cost;
		Not easy to learn;
		Not easy help system;
		Not effective to help users identify online frauds;
		Easy to be replicated

5.0 Conclusions and Future Research

Online attackers take advantage of security vulnerabilities, whilst they often exploit various usability vulnerabilities in authentication mechanisms. Obviously, increasing usability of authentication mechanisms contributes in a remarkable way to the security effectiveness. To improve the usability of different authentication mechanisms, maintaining appropriate authentication strength at the same time, a set of non-biased usability metrics is needed. We proposed a new set of usability metrics to evaluate the authentication mechanisms based on the deductions from the analysis of existing authentication mechanisms and their properties. With the proposed metrics, researchers are able to evaluate different authentication mechanisms and find out how to alleviate their usability weaknesses and, thus, strengthen the service quality. For the designers of the authentication mechanisms, this set of usability metrics can act as a design guideline to help towards the creation of usable and secure online authentication mechanisms.

Our future plan is to include experimentation of the proposed metrics within real use scenarios. First of all, an environment supporting comparisons from usability perspective is needed. Secondly, we only analysed the general aspects of usability issues in authentication mechanisms, but it is not known how to weigh these usability metrics. To get properly weighted values for the metrics, security effectiveness metrics may be needed to mitigate the potential risks and to conclude the trade-offs of security and usability for authentication mechanisms.

6.0 Acknowledgement

We hereby sincerely thank the financial support from Internet of Things research program, issued by Tekes Finland. We also thank Kristian Slavov for his valuable suggestions and comments.

7.0 References

1 Berki E. & Jäkälä M., Cyber-Identities and Social Life in Cyberspace. Hatzipanagos, S. & Warburton, S. (Eds) Social Software and Developing Community Ontologies (London: Information Science Reference, an imprint of IGI Global). pp28-40. London, 2009

2 Dhillon G. & Moores T., (2001). Internet privacy: Interpreting key issues. Information Resources Management Journal, 14, 4, 33-37

3 Warren M., & Hutchinson W., (2002). Cyberspace Ethics and Information Warefare, Social Responsibility in the Information Age: Issues and Controversies. Idea Group Publishing 2001, ISBN-10: 1930708114

4 Wyke J., (2012). What is Zeus, technical paper, http://www.sophos.com/en-us/why-sophos/our-people/technical-papers/what-is-zeus.aspx (visited January 2012)

5 Helenius M., Fighting against Phishing for On-Line Banking Recommendations and Solutions. In Proceedings of the 15th Annual EICAR Conference Security in the Mobile and Networked World", pp252-267, Germany 2006

6 Bellovin S.M., (2004). Spamming, Phishing, Authentication, and Privacy. Communications of ACM 47(12). 144

7 Williamson G. D., (2006). Enhanced Authentication In Online Banking. Journal of Economic Crime Management, 4(2). 2006

8 Ren Q, Mu Y., Susilo W., Mitigating Phishing with ID-based Online/Offline Authentication, proceedings of 6th Australasian Information Security Conference, pp59-64, Australia. 2008

9 Oiwa Y., Takagi H., Watanabe H., Suzuki H., PAKE-based mutual HTTP Authentication for Preventing Phishing Attacks, proceedings of WWW2009 MADRID!, pp1143-1144, Madrid, 2009

10 Tullis T., Albert B., Measuring the user experience: collecting, analyzing, and presenting usability metrics illustrated edition, Focal Press 2008, ISBN-10: 0123735580

11 Savola R. & Abie H., (2010). Development of Measurable Security for a Distributed Messaging System. International Journal on Advances in Security, 2, 4, 2009, ISSN 1942-2636, 358-380

12 163 email, www.163.com, available from May, 2012

13 Nordea online banking, www.nordea.fi, available from May, 2012

14 Alipay (ZhiFuBao), www.alipay.com, available from May, 2012

15 QQ messenger, http://im.qq.com/qq/2012/, available from May, 2012

16 UTA entrance system, www.uta.fi, available from May, 2012

17 Li L. & Helenius M., (2007). Usability Evaluation of Anti-phishing Toolbars, Journal of Computer Virology, 2007, 3, 163-184

18 Villamarín-Salomón R. M., & Brustoloni J. C., (2010). Using reinforcement to strengthen users' secure behaviors, proceedings of the 28th international conference on Human factors in computing systems (CHI '10), pp363-372, New York, NY, USA, 2010

19 Li L., Helenius M., Berki E., (2012). A usability test of whitelist and blacklist-based anti-phishing applications, MindTrek Academic Conference.

20 Gaw S. & Felten E. W., Password management strategies for online accounts, proceedings of the second symposium on Usable privacy and security, Pittsburgh, Pennsylvania, 2006

21 Conlan R. M. & Tarasewich P., Improving interface designs to help users choose better passwords, CHI '06 extended abstracts on Human factors in computing systems, Montréal, Québec, Canada, 2006

22 Leicher A., Schmidt A., Cha I., Shah Y., Smart OpenID Smartcard Webserver Enabled SSO for Web 2.0 using OpenID, presentation slide, http://smartopenid.novalyst.de/wp-content/uploads/2011/05/NOVALYST_InterDigital_Smart_OpenID_presentation_SmartMobility2010_small.pdf, available from June 2012