

NEW APPROACHES FOR ENSURING USER ONLINE PRIVACY

Kaigui Bian

THESIS

submitted to the Faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree

KEYWORDS: online privacy disclosure, phishing attack, child's online privacy, COPPA

Master of Science
in
Computer Engineering

Dr. Jung-Min Park, Chairman

Dr. Y. Thomas Hou

Dr. Yaling Yang

November 26th 2007

BLACKSBURG, VA

© Copyright 2007, Kaigui Bian

New Approaches for Ensuring User Online Privacy

Kaigui Bian

Abstract: With the increase of requesting personal information online, unauthorized disclosure of user privacy is a significant problem faced by today's Internet. As a typical identity theft, phishing usually employs fraudulent emails and spoofed web sites to trick unsuspecting users into divulging their private information. Even legitimate web sites may collect private information from unsophisticated users such as children for commercial purposes without their parents' consent. The Children's Online Privacy Protection Act (COPPA) of 1998 was enacted in reaction to the widespread collection of information from children and subsequent abuses identified by the Federal Trade Commission (FTC). COPPA is aimed at protecting child's privacy by requiring parental consent before collecting information from children under thirteen.

In this thesis, we propose two solutions for ensuring user online privacy. By analyzing common characteristics of phishing pages, we propose a client-side tool, Trident, which works as a browser plug-in for filtering phishes. The experiment results show that Trident can identify 98-99% online and valid phishing pages, as well as automatically validate legitimate pages. To protect child's privacy, we introduce the POCKET (parental online consent on kids' electronic privacy) framework, which is a technically feasible and legally sound solution to enforce COPPA. Parents answer a questionnaire on their privacy requirements and the POCKET user agent generates a privacy preferences file. Meantime, the merchants are required to possess a privacy policy that is authenticated by a trusted third party. Only web sites that possess and adhere to their privacy policies are allowed to collect child's information; web sites whose policies do not match the client's preferences are blocked. POCKET framework incorporates a transaction protocol to secure the data exchange between an authenticated client and a POCKET-compliant merchant.

Contents

Abstract	ii
List of Figures	v
List of Tables	vi
1 Introduction	1
2 Related Work	6
2.1 Anti-Phishing Techniques	6
2.2 Cookie Control	8
2.3 Privacy Policies	9
2.4 Software Controls	11
2.5 Trust Seals	12
3 Motivations	14
3.1 Web-Based Identity Theft	14
3.1.1 Challenges and Requirements	14
3.1.2 New Insights in Client Solutions	16
3.1.3 Feasibility of Implementations	16
3.2 Technical Enforcement of COPPA	17
3.2.1 Available Technology and Seal Solutions	18
3.2.2 Parental Control and Awareness	19
4 Trident: A Client Side Solution to Phishing Attacks	22
4.1 Detection Algorithm of Trident	22
4.1.1 Legitimacy Check	23
4.1.2 Relevancy Check	25
4.1.3 Input Form Check	27
4.1.4 Work Flow of Trident	28

4.2	Implementation of Trident	30
4.2.1	Browser Helper Object	30
4.2.2	Queries in Online Information Repositories	31
4.3	Performance Evaluation	33
4.3.1	Experiment Data Sets	33
4.3.2	Detection against Phishing Pages	34
4.3.3	Validations of Legitimate Sites	36
4.3.3.1	Legitimate Sites	37
4.3.3.2	Legitimate Foreign Sites	39
4.3.4	Comparisons with Other Anti-Phishing Tools	40
4.4	Discussions	41
4.4.1	Manipulation of PageRank	41
4.4.2	Problems in Relevancy Check	42
5	POCKET: Parental Online Consent on Kids' Electronic Transactions	44
5.1	The POCKET Framework	44
5.2	Implementation of POCKET	47
5.2.1	Registration Phase	48
5.2.2	Setup Phase	48
5.2.3	Transaction Phase	49
5.3	Security Features of the POCKET Framework	50
5.3.1	One Ticket Authentication Protocol	53
5.3.2	Registration Phase Ticket Distribution	53
5.3.3	Secure Transaction Phase	53
5.4	POCKET Software Prototype	57
6	Conclusion	60
	Bibliography	62
	Acknowledgements	67

List of Figures

1.1	How phishing evolves from spam.	2
4.1	A backlink tree to a web site.	26
4.2	Trident flowchart.	29
4.3	Trident toolbar and log file.	31
4.4	Miss detection rate of Trident when detecting phishing pages.	35
4.5	False alarms when evaluating legitimate web sites.	38
4.6	False alarm rate when evaluating legitimate UK sites by varying <i>PRth</i>	39
5.1	Example POCKET preferences XML File.	45
5.2	The POCKET registration and setup phase.	46
5.3	The POCKET transaction phase.	49
5.4	Building the POCKET installation package.	52
5.5	Transaction phase handshake protocol.	54
5.6	Parent creating the UPPF for mike allowing only First Name, Age Range and Zip Code to be collected.	56
5.7	Status dialogs for the POCKET user agent.	57

List of Tables

4.1	Search Depth when a phish is detected by Trident.	36
4.2	Number of phishes detected by Trident modules.	36
4.3	Search Depth when a legitimate site is verified by Trident.	37
4.4	Proportion of sites verified by legitimate backlinks.	37
4.5	Search Depth when a legitimate UK site is verified by Trident.	40
4.6	Proportion of UK sites verified by legitimate backlinks.	40
5.1	Legend of Terms Used	51

Chapter 1

Introduction

A paramount concern of individuals using the Internet is the protection of their privacy. Recent surveys show that the awareness regarding personal privacy is growing. A Harris poll [15] designed by Privacy & American Business and sponsored by Microsoft found that 35% of Americans had “very high privacy concern.” 65% had refused to register at an e-commerce site because of privacy concerns while, 60% decided not to patronize a site due to doubts about the company’s policies. 7% had filed a complaint regarding misuse of personal information.

Online financial crimes caused by identity theft draw the public attention in recent years. Spam techniques make it easy to indiscriminately send unsolicited bulk messages to numerous recipients [58]. Phishing is a deceptive attack distributed via spam messages, which allures recipients into giving out private information on spoofed (phishing) web sites or pages. Phishing web sites are slightly different from spam web sites since phishing web sites require users’ private information, and they are always short-lived (typically a few hours) [28]. Fig. 1.1 depicts how phishing evolves from spam. Phishing attack has been

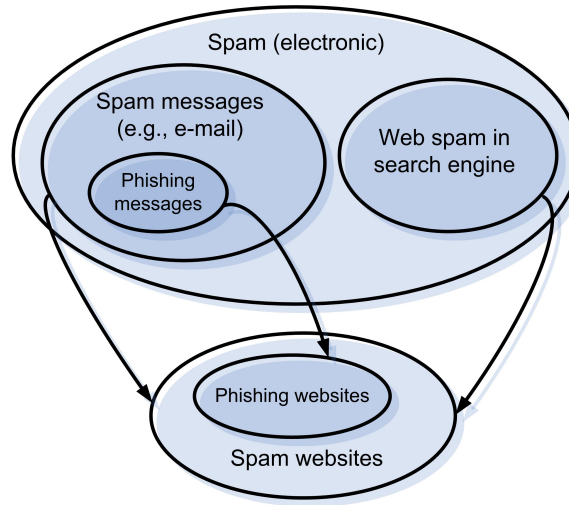


Figure 1.1: How phishing evolves from spam.

so successful to cause enormous financial loss every year. The Gartner group estimates that the direct phishing-related loss to U.S. banks and credit card issuers in 2003 was \$1.2 billion [16]. Data suggests that some phishing attacks have convinced up to 5% of their recipients to provide sensitive information to spoofed web sites [61].

Existing anti-phishing schemes have several limitations:

- Most detection algorithms against phishing rely on the content analysis of the suspected page [6] [6] [41] [61], or a blacklist that includes the urls or domains of known phishing web sites [6] [22] [30]. Phishers can circumvent the content analysis detection algorithm by evolving their page spoofing techniques.
- Some anti-phishing tools focus on preventing any unauthentic user input. Once an unverified web input form is detected, the users will be forced to take the suggested countermeasures [6] [61] [41]. Studies [60] show that users feel so disturbed by unnecessary interruptions when they visit legitimate sites.

- Most of anti-phishing tools require storing data locally at the client side, such as a verified white list, or personal login card, which may be compromised by adversaries' sniffing and modifications [24].

With the increasing use of the Internet for commercial purposes, Children's privacy has never been so critical. 65% of children between the ages of 10 and 13 use the Internet, and some sites ask for their personal information for commercial purposes without the parental consent. Recognizing the importance of protecting children's privacy on the Internet, the Children's Online Privacy Protection Act of 1998 (COPPA) requires parental consent before web sites can collect information from children under the age of thirteen. The FTC has adopted regulations to enforce COPPA. Unfortunately, the current business practices used and the technical approaches employed to comply with COPPA, fail to protect children's privacy effectively. The present research addresses this issue by: (1) evaluating the awareness and use of privacy protection tools by parents, and (2) developing a tool to provide or deny parental consent for online collection of information from children.

In our research on phishing attacks, we found that most phishing pages are created on insecure web sites owned by phishers. These insecure sites have few (high quality or high importance) inbound links (a.k.a. backlinks or inlinks). The quality or importance of a link is quantified by Google PageRank [38] score. A page with very high PageRank score is considered legitimate. In some rare cases, the PageRank algorithm may be manipulated, which will be discussed later. Another finding is that some phishers can break into legitimate sites and created phishing pages on those sites. These phishes can evade the detection of blacklist and domain check. However, these phishing pages often have irrelevant topics to the legitimate host sites. These findings can be used for detecting phishing pages.

From the preliminary results of the parental awareness studies we find that very few parents are aware of the laws that exist to protect children. In addition to the lack of awareness, technical issues need to be addressed. Verifiable parental consent is one factor of COPPA that, technologically, has proven to be one of the most challenging to implement. Web sites are allowed to decide the method they use for obtaining parental consent for collecting the child's personal information [19]. The FTC hoped that technological advances will provide a more sophisticated, reliable, and cost-efficient manner to obtain consent.

In this thesis, we first propose "Trident", a client anti-phishing tool (browser plug-in) that filters phishing pages. Instead of analyzing the phishing page itself, Trident performs a legitimacy check on the backlinks to the page's host site; it conducts a relevancy check on the web categories of the page and its host site; and it carries out an input form check to confirm the suspected page really collects users' information. A web site is legitimate if its backlinks are trusted; a web page on a legitimate site is trusted if its web category is consistent with its host site's web category. All the data used by Trident can be obtained from reliable third parties (repositories) which eliminate the need for any local data storage. The impacts of Trident are of three-fold:

- It can differentiate less known but legitimate web sites from phishing web sites.
- It can detect newly emerging phishing pages that break into legitimate hosts.
- It does not require storing any data at client side, which minimizes the risk of being compromised by adversaries.

In addition to client side phishing filter, we also develop a tool that provides a reliable, trustworthy technology option for obtaining verifiable parental consent as required by COPPA. The tool's effectiveness will depend upon meeting parental acceptance and business standards. The proposed tool, called POCKET (parental online consent on kids' electronic privacy) is described further in this thesis. POCKET provides an easy-to-use interface for parents to configure privacy choices for their child, and then automatically enforces these policies.

The rest of this thesis is organized as follows. In Chapter 2, we introduce related work. Motivations of our work are presented in Chapter 3, followed by the client side anti-phishing tool Trident in Chapter 4. In Chapter 5, we provide details of POCKET framework. Finally, we conclude our work in Chapter 6.

Chapter 2

Related Work

In this chapter, we summarize existing techniques that have been proposed and used to protect consumer privacy online.

2.1 Anti-Phishing Techniques

Today's detection algorithms against spoofed or phishing pages employ heuristic rules for evaluating the page content. SpoofGuard [6] applies multiple tests on the suspected page and combine these test results using a scoring mechanism, which yields a total spoof score for determining the spoofed page. Other proposed solutions also depend on content analysis. For example, web wallet [61], dynamic skin [13] and iTrustPage [41] first analyze the HTML source code to seek suspected input forms, and then prevent users from disclosing credential information in that input form. Such an input form check cannot exactly distinguish legitimate login pages from phishing pages. If a login form appears in a less known

login page that has not been verified, users will be forced to take suggested countermeasures that may be intrusive or annoying. Similarly, Microsoft phishing filter [30] analyzes pages you are browsing and determine if they have any characteristics that might be suspicious. It also checks the sites you are visiting against an up-to-the-hour, dynamic blacklist of reported phishing sites. FireFox phishing protection tool [22] also employs a blacklist to check the history of suspicious pages. Both of them fall into the category of content analysis (url and domain checks). The accuracy of such detection algorithms is somewhat limited by the updating frequency and coverage of the blacklist. When evaluating emerging spoofed pages that break into legitimate sites, miss detection will happen.

Most anti-phishing tools work as the browser plug-in. SpoofGuard is such a browser plug-in that can generate pop-up warning messages, and give different colors to indicate the risk levels in its toolbar. Microsoft phishing filter is another browser plug-in embedded in IE. If it finds suspicious characteristics in web pages, it will show a yellow warning, advising you to proceed with caution. If it finds a match in the blacklist, it will show you a red warning notifying you that the site has been blocked for your safety. FireFox phishing protector is a FireFox browser plug-in that reacts in a similar way as Microsoft phishing filter. Other tools provide toolbars that prevent users from directly inputting private data into spoofed login forms. Web Wallet provides an additional browser sidebar. If the current page is safe, Web Wallet will fill the input form for users by using local login card. If not, web wallet will ask for which site the user intends to access. The user will be given an alternative safe path to her intended site. iTrustPage uses very conservative rules to evaluate the login page. If the login page has input forms, the browser will redirect the users to a warning page for searching the intended page. It may be too intrusive and

annoying to users by redirecting them.

In addition to above client side tools, some other solutions require server side supports. In dynamic skin, the client tool will generate a visual background of password window for users, and the server will also generate a visual border for password window. Users only need to recognize their personal image and perform one visual matching to compare two images. Similarly, Bank of America's SiteKey [1] is another example of site authentication image check. The bank users can verify the SiteKey (site-authentication image) before entering the password. Another example of site-authentication image is Yahoo! sign-in seal [62]. By placing cookies and/or Macromedia Flash objects on the computer Yahoo! associates site-authentication images with computers, rather than individual user accounts (what BoA does). All users of a computer share the same site-authentication image. The users of a computer can verify the Yahoo! sign-in seal (image) before entering the "Yahoo! ID" and the password. Although the involvement of server support will make these tools more reliable, a survey [42] shows that most users may ignore the site authentication image and continue disclosing personal information, which raises challenges for wide deployment on all servers. In addition, requirements of server support may hinder the wide deployment of the scheme.

A more comprehensive survey about anti-phishing techniques is available in [16].

2.2 Cookie Control

The advent of *cookies* as an extension to the stateless HTTP protocol standard allowed web sites to tag a browser with information that would be available to the server when the user returns. Cookies provide a method to track the visitor to a website, and websites

started storing this tracking information for extended periods. The use of cookies raise privacy concerns [25] when third party cookies started linking collected browsing history with other gathered information [14].

To protect user privacy, browser companies including Microsoft and Netscape provide privacy control features in browsers to limit the collection of information through cookies. Anonymizers offer anonymous web surfing by acting as an intermediary between the user and the website. Most anonymizers prevent the website from tracing the client's IP address or placing cookies in the viewer's computer. These are true privacy-enhancing technologies [14], because they remove identifying information completely.

2.3 Privacy Policies

Companies and websites may use a privacy policy to provide notice to consumers of the websites' information collection practices. The privacy policies may outline the kind of information being collected, the purpose for collecting the data, companies with which this information is shared, whether the consumer has access to the data, online contact information, etc. Generally speaking, outside of COPPA and topic specific laws, there is no requirement for a website to post a privacy policy.

Legal and privacy experts prepare the policies which include terminologies that make it difficult for ordinary consumers to read and understand. The Platform for Privacy Preferences project (P3P) created by the World Wide Web Consortium (W3C) was aimed at creating a machine readable, common vocabulary for identifying privacy practices. The P3P policies are expressed in the eXtensible Markup Language [56] (XML)¹ format. P3P

¹XML is a W3C language for general-purpose markup.

enables the websites to express their privacy policies in a standardized machine readable format so that automated tools (or user agents) can interpret them [10]. P3P defines eight major components containing multiple subcomponents and attributes. It defines a number of purpose subelements that define each category for data use.

P3P also includes syntax for compact policies² to represent the site's data practices for cookies. This feature is used by several existing browsers to make decisions regarding blocking or allowing cookies [9]. The Internet Explorer 6 (IE6) web browser implements a P3P based cookie management system [31]. The IE6 privacy features filter cookies using these compact policies based on the user's privacy settings. IE6 allows users to perform coarse cookie control by selecting from among six different preconfigured settings (from *block all* to *allow all*). Several other browsers also have similar privacy controls based on compact P3P.

Many P3P user agents were developed during the standardization process of P3P. These user agents inform the consumer of the site's practices so that the user can make an informed decision regarding using a particular website. Privacy Bird [11] is the most advanced, easy to use and open-source user agent implemented at the AT&T Labs for visualizing the privacy policies. The Privacy Bird displays an icon of a bird that changes color, and provides a vocal feedback when the preferences of the client and website policies match (differ). The user has the option to continue (stop) using the website.

²The elements of the P3P policy are mapped to short tokens and aggregated to form a compact policy.

2.4 Software Controls

Several parental control packages that protect children from inappropriate content on the Internet are available today. Tools include browser add-on modules, dedicated software, and operating system features that help prevent children from accessing such materials online. These tools can be used to monitor the child's activities on the computer and Internet, but do not address privacy issues. Microsoft's new operating system Windows Vista provides more advanced parental control features that can be used along with the Internet Explorer 7 (IE7) [29]. Parents can restrict their children to playing particular games, running specific programs and visiting specific websites. In addition, parents can use Vista to configure time limits on the child's daily computer use.

Net Nanny [36] is a parental control software that allows parents to monitor the child's activity-logs, set time limits on computer use and block access to certain software. Net Nanny 5 also blocks sending a set of configurable private information in outbound communication. The ParentalControl Toolbar [59] is another privacy control feature available as an extension to various browsers. It helps parents prevent the children from viewing adult-oriented websites. The toolbar assumes that websites voluntarily label the pages based on the Internet Content Rating Association's (ICRA) vocabulary. The ParentalControl Toolbar uses these labels to decide whether the website contains suitable content and blocks websites containing inappropriate content.

2.5 Trust Seals

In an attempt to self-regulate privacy concerns in general, industry groups developed trust “seals” targeted at reassuring consumers that the companies displaying the seals abide by the seal program’s privacy rules. The major programs include TRUSTe [53], BBBOnline (Privacy) [2], and the CPA WebTrust [7]. Similarly, most current solutions for meeting the requirements of COPPA primarily have been based on seal programs rather than on technical solutions. These programs generally comprise a standard agreement by a website to protect children’s privacy, the payment of a fee, routine audits, and an online dispute resolution process. Upon completion of this process, the website is allowed to post a children’s privacy seal, which indicates to the users that the site is compliant and certified. In theory, seals are designed to induce brand-like recognition and stimulate trust in the website, therefore increasing the chance that a person will use the site.

Researchers have studied the effectiveness of seal programs only at the adult level. Conflicting evidence muddles our understanding of the effectiveness of general trust seals. In a 2001 study, Harris reported that when consumers notice privacy seals they consider them important and are more willing to provide personal information to the site because of the third party verification [49]. Recently, a survey conducted for Privacy and American Business found that 91% of consumers would feel more comfortable using sites participating in a third party verification program and 84% believed that they should be required for electronic businesses. Furthermore, 62% of respondents believed that third party privacy seals would reduce their privacy concerns [37]. However, other evidence suggests that consumers in general and even experienced web users may not recognize privacy

and security seals [48]. Only 25% of consumers seem to recognize seal features on Web sites [49]. Findings also demonstrate that online shoppers view trust seals as less important than other factors such as reputation and convenience [4], although trustworthiness of the merchant and system affect users' intentions [27, 44]. These findings are consistent with the results of a previous study where fewer than 14% of 2,000 experienced users said they would trust a site that has a third party seal [26]. Given these results, we have little reason to believe that a child privacy seal is an effective tool for notifying the users of the trustworthiness of a website.

Chapter 3

Motivations

3.1 Web-Based Identity Theft

Phishing attacks are proliferating on internet today, and sophisticated phishers make their scam pages difficult to detect by evolving their obfuscation techniques. As a result, existing anti-phishing client tools based on content analysis have difficulties in differentiating phishes from legitimate sites and pages accurately. Before introducing Trident, we present three challenges faced by the security design of a client anti-phishing solution.

3.1.1 Challenges and Requirements

In general cases of phishing attacks, new phishers create their own sites with a batch of spoofed pages. The urls of these phishing pages have a typical format as,

www.newphish.com/.../phishpage.

The urls of web pages under a less known but legitimate web site (e.g., a foreign web site) take the similar format as,

www.foreignsite.com/.../foreignpage.

Since most phishing sites are short-lived, such a new phishing site *www.newphishsite.com* is always unfamiliar or unknown to public, so is a less known web site *www.foreignsite.com* from an unfamiliar domain. Many anti-phishing tools mistakenly determine a less known web site with login forms as a phish. Hence, it is a challenge to differentiate such a new but fraudulent site from a less known but legitimate site.

Another problem is that adversaries can break into a legitimate website, or some legitimate sites are accessible to anonyms. Those sites that cannot be compromised or modified by anonyms are called *inaccessible* sites. It is possible for phishers to graft spoofed pages on legitimate but accessible sites. The url of such a phishing page has a typical format as,

www.legitimate.com/.../phishpage.

Many anti-phishing tools (e.g., blacklisting and domain check) miss detecting such a phishing page, because the host site is legitimate, and the phishing url is similar to that of other legitimate pages on the same site, such as

www.legitimate.com/.../legitimatepage.

Here, another challenge is raised for detecting phishing pages that break into accessible legitimate web sites.

Thirdly, some anti-phishing tools employ a pre-verified white-list, or personal login cards, or personal images, which are stored in cookies or local files at the client side.

However, these local data may raise new security concerns since they are vulnerable to browser sniffing or malicious modifications. Adversaries may succeed in eavesdropping or modifying local data to circumvent the anti-phishing tools. To minimize the potential security risks, it is better to make the client solution independent of local stored data.

3.1.2 New Insights in Client Solutions

Intuitively, the best way to assess a new friend may not be to judge him by his appearance, but to see if he has reputable friends. If he has reputable friends, this new friend could be considered as reliable. In another example, to evaluate the work of a friend, it is better to first check if the work is within his specialty. If the work is out of his expertise, the work quality might be expected as dissatisfied.

Analogically, we trust a less known web site if it has friend links with high reputation. Otherwise, this site may be a new or short-lived site which implies a risk of phishing. If an accessible legitimate web site has pages with irrelevant topics (web categories), these pages may be created by intruders. Thus, friend links and web categories can help detection against phishing sites and pages. To minimize the need for storing data at the client side, the ideal way is to obtain all data needed (e.g., friend links and web categories) from external online repositories that are reliable and accessible. Next, we explore the feasibility of implementing these ideas.

3.1.3 Feasibility of Implementations

First, we need a method to quantify the legitimacy of a web site. Given a web site's url, a score in the range of $[0, 10]$ could be generated to evaluate the site by some reliable third

parties, such as Google and TrustGauge. Google PageRank score indicates the importance and reputation of a site, while TrustGauge score [19] measures the trustworthiness of a site. In Trident, we prefer PageRank score since Google has more reliable and comprehensive database than TrustGauge. If its PageRank score is higher than a threshold, we consider the web site as legitimate.

Friend links to a site are also known as inbound links or backlinks or inlinks that link from other pages to that site. Either Yahoo! or Google or msn search supports “link:url” commands for queries on backlinks to a url via web interfaces. Yahoo! site developer API [63] also provides inlink data for supporting applications. With Yahoo! backlink service, we can constitute a backlink tree to a web site with this site as the root, in which Trident searches for any legitimate backlink.

To evaluate pages on an accessible legitimate site, the PageRank score and backlinks may not be sufficient since many legitimate pages have no backlinks or low PageRank scores. It is difficult to distinguish these pages from spoofed pages created by intruders in this site. Instead, we need to check the web topic relevancy between the evaluated page and its host site. Major search engines provide web directory service, which return pages in categories relevant to the given keywords. It is feasible to get the web category of a page by giving its keywords to web directory service. Then, we are able to check the web category relevancy of the page and its host site by comparing their categories.

3.2 Technical Enforcement of COPPA

As described in the review of related technology, there is no widely available, effective technology that implements a parent’s ability to choose what information is shared by

a child with a commercial web site. In addition, while children need protection online from the dangers of sharing personally identifiable information because they are socially immature and naïve, they are correspondingly sophisticated about the use of the Internet and technology. Computers and Internet usage are ubiquitous, frustrating the busy and less technological savvy parent from fully protecting their child online. A recent example of this problem is the case of Xanga.com [21], an interactive social networking site that was fined \$1 million dollars by the FTC for failing to effectively implement parental consent for children to use the site. Its failure was massive, with over 7 million children accessing the site, created profiles with birth dates indicating they were under 13. Further, Xanga.com failed to notify parents about their information collecting practices or provide access and control to the information collected from children. Social networking sites, where personal information abounds, can pose a special danger to children who may share offline identifying information that will allow them to be contacted or tracked. By implementing parental control over the personally identifiable information that a child can share, COPPA intends to empower parents to protect their children. Yet, as the Xanga.com case shows, the protocol as it now exists requires a web site to contact the parent for consent, and children are adept at circumventing web site procedures. POCKET is designed to reverse the procedure and will allow a parent to control access unless the web site consents to the information collection parameters.

3.2.1 Available Technology and Seal Solutions

As outlined in Chapter 2, P3P is a system that was meant to specify a privacy policy in well defined language constructs so that automatic negotiation of privacy preferences

could take place and indicate when privacy mismatches occurred. However, research into the implementation of P3P found errors in policy implementation, violations of policy, and incomplete policies [8], leading the authors to recommend the use of a third party to certify compliance. Even if P3P could be implemented in a dependable manner, the privacy language constructs are not fine grained enough to address the specific choices that a parent would make regarding the exact information that a web site could or could not collect or share. Lastly, and most importantly, P3P does not incorporate any method whatsoever for inserting the parent's consent over the child's information; it is a two party negotiation system only.

Privacy seal programs have been implemented in order to address concerns that web sites were not honestly revealing their practices to consumers, and that lack of privacy was discouraging consumers from participating in electronic commerce. However, researchers have also found that privacy seals in general do not engender trust from online consumers [4]. Because privacy seals in general have not been successful, it is unlikely that COPPA compliant seals will fare any better.

3.2.2 Parental Control and Awareness

COPPA seeks to protect children by giving parents control over what information their children can share. For parents to protect their children, they must be aware of web site information collection practices and their right and ability to control this. We used a focus group approach to understand privacy protection awareness, elicit requirements for POCKET, and receive feedback on the prototype.

For these focus groups a protocol was developed, tested, and modified several times.

Parents were invited to participate in one of four focus groups without knowledge of focus group details (to avoid self selection bias). The session lasted 60 minutes. Each focus group had between 3 and 6 people who ranged in age from 29 to 48 years old with children under the age of thirteen: for a total of 14 females and 4 males. Eleven browsed the web everyday, two several times a week, four once a week, and one once or twice a month. Five of the parents are high school graduates, three graduated from a two-year college, five have bachelor's degrees and five have graduate degrees. Twelve of the parents' children browsed the web more than several times a week, three once a week, two once a month, and one never. Two researchers attended and moderated the focus groups, which were recorded.

Results from these focus groups indicate that while most parents warn children not to give information online, they are not sure children respect that; one of them knew of laws protecting children's information online; none of them knew if merchants do anything to protect children's privacy online; two had been asked for their consent by a web site for their children's information; and, none knew where to report fraudulent web activity. None had ever "heard of" Children's Advertising Review Unit (CARU) [5]. All but one had "never heard of" PRIVacy Vaults Online (PRIVO) [40], Entertainment Software Rating Board (ESRB) [17], or TRUSTe [53]; one had "vaguely heard of it." Ways parents use to control what children view online include placing stations in plain view; telling children they could track sites browsed (though admitting they didn't know how); one uses the AOL age block and tracking service; and one uses software called "Content Protect."

Looking at these results we strongly feel that the POCKET framework is the best solution. It is an easy to use, automated tool that can be used by a technologically unsophisticated parent and deployed to protect children's privacy. It puts control into the hands of the parent and identifies the web site visitor as a child. POCKET allows parents to implement the choice that NO information be collected from their child. Automation provides the advantage that the parents do not have to constantly supervise and worry about disclosure of personal information by their children. Once deployed it will provide a way to enforce the parental consent requirement of COPPA. In the next few sections we discuss more details regarding the POCKET framework and the security protocols involved to ensure children's privacy.

Chapter 4

Trident: A Client Side Solution to Phishing Attacks

4.1 Detection Algorithm of Trident

In this section, we present the detection algorithm of Trident against phishing pages, which includes three modules: legitimacy check, relevancy check and input form check. First, we formulate the web as a directed graph $G = (V, E, L)$, where V is the set of web sites (vertices), E is the set of web links, and L is the legitimacy tags of vertices in V . If there is a link from site u pointing to site v , we have $e(u, v) = 1$; otherwise $e(u, v) = 0$. If $u \in V$ and $l(u) = 1$, it means the site u is validated as legitimate; otherwise the site is unverified ($l(u) = 0$).

4.1.1 Legitimacy Check

In a practical scenario, the web master of a legitimate web site examines the legitimacy of a friend site before exchanging links with that site. Some web masters may use the PageRank score as a general measurement of the reputation of a site. If a friend site has a high PageRank score, it could be linked by other sites. Thus, we have a *legitimacy transitive rule* as:

Proposition 1. *Suppose that two sites $u, v \in V$, $e(u, v) = 1$ and $l(u) = 1$. Then, $l(v) = 1$.*

Given a test page W , its host site H . We construct the backlink tree of H as $T(H) = (V_{T(H)}, E_{T(H)}, L_{T(H)})$, which is a sub-graph of G . The root of the backlink tree is the host site H . Denote the maximum depth of the tree as d , the breadth of the tree (the number of backlinks we collect at each level) as r . The backlinks to site H are defined as the 1^{st} level backlinks of H in the backlink tree; the backlinks to one of the 1^{st} level backlinks of H are defined as the 2^{nd} level backlinks of H in the backlink tree; and the i^{th} level backlinks of H in the tree can be defined similarly. Denote the i^{th} level backlinks of H as $\mathbf{B}_i(H) = \{b_{i,1}, b_{i,2}, \dots\}$. Let $x_0 = H$, and $x_i = \operatorname{argmax}_{j \in \mathbf{B}_1(x_{i-1})} \{PageRank(j)\}$, where $i = 1, 2, \dots, d$. Thus, we have

$$\begin{aligned} V_{T(H)} &= (\mathbf{B}_1(H), \mathbf{B}_2(H), \dots, \mathbf{B}_d(H)) \\ &= (\mathbf{B}_1(x_0), \mathbf{B}_1(x_1), \dots, \mathbf{B}_1(x_{d-1})) \\ &= \left\{ \begin{array}{cccc} b_{1,1} & b_{2,1} & \dots & b_{d,1} \\ b_{1,2} & b_{2,2} & \dots & b_{d,2} \\ \dots & \dots & \dots & \dots \\ b_{1,r} & b_{2,r} & \dots & b_{d,r} \end{array} \right\}. \end{aligned}$$

In the link set $E_{T(H)}$

$$e(u, v) = \begin{cases} 1 & \text{if } u \in \mathbf{B}_1(v); \\ 0 & \text{otherwise.} \end{cases}$$

Initially, all sites in $T(H)$ are unverified, thus $L_{T(H)} = \mathbf{0}$.

In addition, a threshold $PRth$ is defined to determine a site as legitimate or unverified. If $u \in V_{T(H)}$, $PageRank(u) \geq PRth$, the site u is determined as *legitimate* and we let $l(u) = 1$; otherwise, the site is unverified and $l(u) = 0$.

Lemma 2. *If $b_{i,j} \in V_{T(H)}$ and $l(b_{i,j}) = 1$, then $l(H) = 1$. The site H is legitimate.*

Proof. In $V_{T(H)}$, $b_{i,j} \in \mathbf{B}_j(H)$, i.e., $b_{i,j} \in \mathbf{B}_1(x_{j-1})$, where $1 \leq i \leq r$, $1 \leq j \leq d$. Then, in $E_{T(H)}$, we have $e(b_{i,j}, x_{j-1}) = 1$, $e(x_{j-1}, x_{j-2}) = 1$, ..., $e(x_1, H) = 1$.

According to the legitimacy transitive rule (*Proposition 1*), we have $l(x_{j-1}) = 1$, $l(x_{j-2})$, ..., $l(x_1) = 1$, and $l(H) = 1$. Thus, the site H is legitimate. $\square$

The legitimacy check algorithm is given by *Algorithm 1*.

Lemma 3. *The time complexity of legitimacy check algorithm has an upper bound rd .*

Proof. Since in the worst case legitimacy check algorithm has to consider all sites in the backlink tree, the time complexity is no more than $|V_{T(H)}| = rd$. $\square$

Fig. 4.1 depicts an example of a backlink tree of site H . Suppose site H has a PageRank score of 3, which is below a predefined threshold $PRth = 5$. Thus, we need to check its backlink tree. In this example, $d = 3$ and $r = 3$. None of its 1st level backlinks has a high PageRank score, we further check the 2nd level backlinks of site H . We select one of the 1st level backlinks with the highest PageRank score (node C) and check its 1st level backlinks. In Fig. 4.1, site H is determined as legitimate because it has a legitimate

Algorithm 1 Legitimacy Check

Input: A backlink tree graph of site H , PageRank threshold $PRth$.

Output: $l(H)$.

```
1: if  $PageRank(H) \geq PRth$  then
2:    $l(H) = 1$ , goto 16
3: end if
4: for  $j = 1$  to  $d$  do
5:   for  $i = 1$  to  $r$  do
6:     if  $PageRank(b_{i,j}) \geq PRth$  then
7:        $l(b_{i,j}) = 1$ 
8:       for  $k = j - 1$  to  $0$  do
9:          $l(x_k) = 1$ 
10:      end for
11:     else
12:        $l(b_{i,j}) = 0$ 
13:     end if
14:   end for
15: end for
16: print The decision of legitimacy check on  $W$  is  $l(x_0)$ .
```

backlink E (PageRank score $\geq PRth$) in the backlink tree by following the chain of links $E \rightarrow C$ and $C \rightarrow H$.

4.1.2 Relevancy Check

If $PageRank(H) < PRth$ but $l(H) = 1$, Trident performs a relevancy check between H and W to filter phishing pages that break into this accessible site. Denote keywords of W as K_W , keywords of H as K_H , the category of keywords K_W as $C(K_W)$, and the category of page W as C_W .

In Trident, we simply equate two concepts: the category of a page and the category of its keywords, i.e. $C(K_W) = C_W$ and $C(K_H) = C_H$. If $C(K_W) = C(K_H)$, then $C_W = C_H$ and page W is determined as a *legitimate page*; otherwise, it is an *unverified*

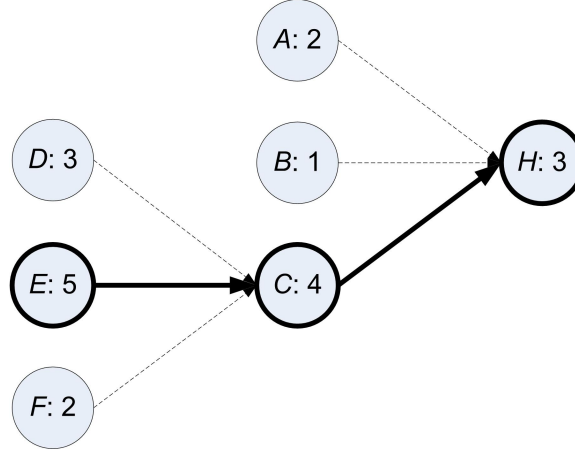


Figure 4.1: A backlink tree to a web site.

page which has a web category (topic) irrelevant to its host site. The algorithm of relevancy check is given by *Algorithm 2*.

Algorithm 2 Relevancy Check

Input: A test page W . **Output:** A decision of topic relevancy check.

- 1: $K_W \leftarrow \text{Keyword}(W), K_H \leftarrow \text{Keyword}(H)$,
 - 2: $C_W \leftarrow \text{Category}(K_W), C_H \leftarrow \text{Category}(K_H)$
 - 3: **if** $C_W == C_H$ **then**
 - 4: **print** The page W is legitimate since it is relevant to its host site H .
 - 5: **else**
 - 6: **print** The page W is unverified since it is irrelevant to its host site H .
 - 7: **end if**
-

Trident employs *SEO* (search engine optimizers) keyword extraction tool [43] for obtaining keywords of the web page. It accepts a url as input, and returns the keywords of the page ranked by frequency. We pick up the top three English keywords, each of which has more than two letters. Yahoo! web directory provides 14 top web categories, and each top category has a number of 2^{nd} , 3^{rd} , or n^{th} level sub-categories. From a top category to one of its n^{th} level sub-category, we obtain a category path. Given query keywords,

Yahoo! directory search returns web pages relevant to the keywords, and each of them has a full category path. For example, if you make queries with keywords “web search”, the first result is “Yahoo! Search” with a full category path as, *Computers and Internet > Internet > World Wide Web > Searching the Web > Search Engines and Directories > Yahoo!*. The top category of keywords “web search” is “*Computers and Internet*”. We categorize these 14 top categories as commercial and non-commercial, which is a binary classification method. Denote N as the number of categories in the classification method, and we have two options: $N = 2$, or 14. In either method, a category is expressed by an integer from 1 to N .

4.1.3 Input Form Check

If the test page W is determined as an unverified page by relevancy check or its host site H is determined as an unverified site by legitimacy check, Trident conducts a third check to further confirm this page is a phish that requests personal information or an unverified page without any misbehavior.

Similar to content analysis solutions, Trident searches the HTML source code for an input form. An unverified page is determined as a phish if it has these features: an input form and the “action” field of this form is “submit”. These features imply that the user input will be forwarded to the back end web server (phishers). If these features are not satisfied, this page is simply an unverified without any misbehavior regarding privacy disclosure. At the end of an input form check, Trident will notify the user of our decision if it is a phish or an unverified page.

4.1.4 Work Flow of Trident

The brief work flow of Trident is as follows:

1. Given a test page W , Trident first checks the legitimacy of its host site H . If the host site has a PageRank score at least PR_{th} , the test page is determined as legitimate.
2. If the host site is determined legitimate but may be accessible, Trident performs a relevancy check on the web categories of W and H . If these categories are consistent, the test page is determined as legitimate. Otherwise, page W is determined as unverified.
3. If the test page or its host site is unverified, Trident performs a third check. If an input form is detected, page W is determined as a phish. Otherwise, it is determined as an unverified page.

The flowchart of Trident is shown in Fig. 4.2.

Theorem 4. *The time complexity of Trident detection algorithm has an upper bound as $(rd + 2)$.*

Proof. In the worst case legitimacy check algorithm has to consider all sites in the backlink tree. Relevancy check and input form check are two one-time operations, and thus the time complexity of Trident detection algorithm is no more than $(rd + 2)$. $\square$

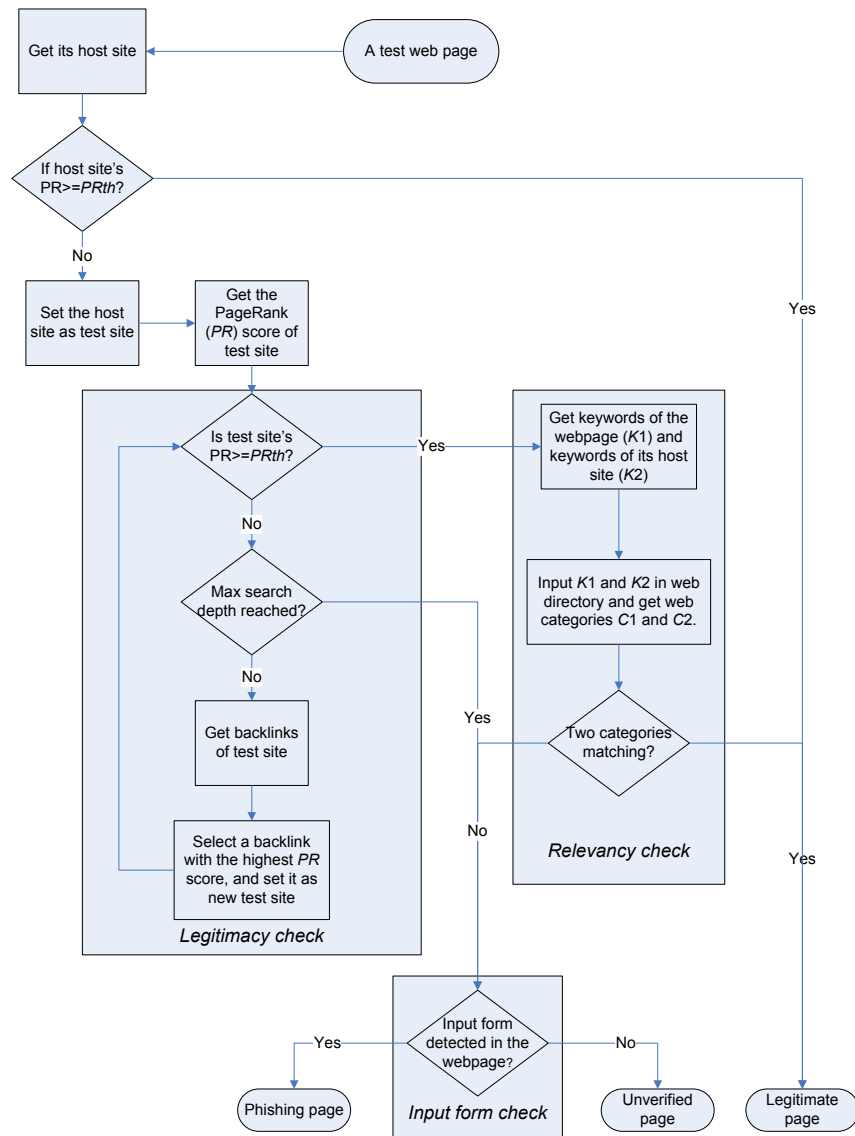


Figure 4.2: Trident flowchart.

4.2 Implementation of Trident

Using C++, we implement Trident as a browser plug-in (browser helper object) embedded in Microsoft Internet Explorer, which protects user online privacy by filtering phishing pages and giving warning messages to unverified pages.

4.2.1 Browser Helper Object

A browser helper object (BHO) is a COM component that is loaded when IE starts up [18]. Trident is a BHO that runs in the same memory context as the browser. In general, a BHO can perform any action on IE windows and modules including manipulating the browser menu and toolbar, detecting and responding to browser events, and creating additional windows. Trident captures the browser event *BeforeNavigate2*, gets the url string from the browser object *WebBrowser2*, and then evaluates the page specified by that url string before displaying the page.

In addition to a COM component, Trident also extends *IDeskband*—an interface that make IE load Trident’s registered toolbar, and the appearance of Trident toolbar is defined by another interface *CWindowImpl*. The toolbar is shown in Fig. 4.3. If users click the button in the toolbar, the Trident log file will be displayed to users. If a page is identified as phish or unverified, a pop-up warning message is shown to users.

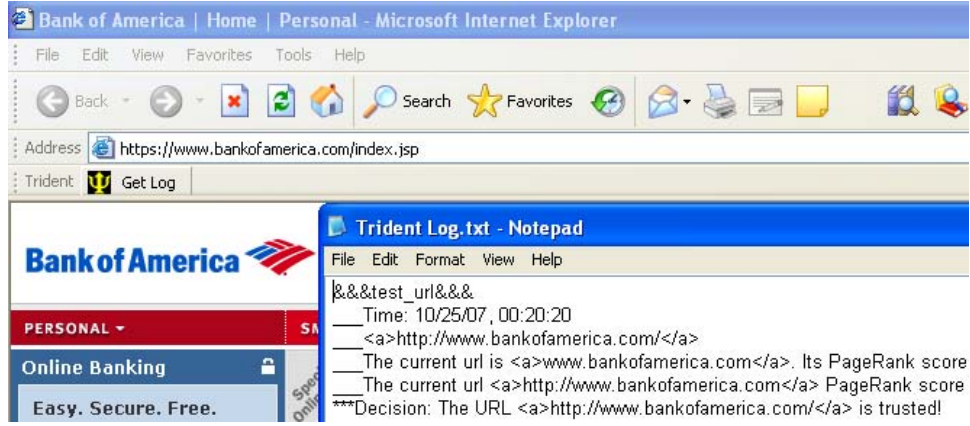


Figure 4.3: Trident toolbar and log file.

4.2.2 Queries in Online Information Repositories

The *WinInet* API enables users to access resources on the World Wide Web (WWW) [35]. External online database servers accept WWW queries so that Trident can establish connections to these servers via the http protocol.

Before starting an http connection with a server, Trident calls *InternetOpen* that gets the root *HINTERNET* handle. Then, it calls *InternetConnect* function to start a connection with the server. A WWW query is made by an http GET request. By calling functions *HttpOpenRequest* and *HttpSendRequest*, it sends the *GET* request with query commands. The query command is contained in the parameter *LPCTSTR lpszObjectName* when calling the *HttpOpenRequest* function.

In legitimacy check, if Trident queries for the PageRank score of *www.google.com*, it composes a query command in the *GET* request as:

/search?client = navclient - auto&ch = 6340563836&features = Rank&q =

info : www.google.com. (4.1)

The string “6340563836” is transformed from the url *www.google.com* by a transformation function that accepts a url as input and outputs a string as the query command. The returned results of *GET* request can be read via the method *InternetReadFile*. By analyzing the content downloaded by *InternetReadFile*, we can get the string that contains the PageRank score.

When making queries on Yahoo! backlinks, another query command is sent via *GET* request to Yahoo! site explorer server *search.yahooapis.com*. The query command is in the format of:

$$\begin{aligned} & /SiteExplorerService/V1/inlinkData?appid = YahooDemo\&query = \\ & \quad www.google.com\&result = 2. \end{aligned} \quad (4.2)$$

The string “*inlinkData*” specifies that the service type is inlink (backlink) query, and the string “*&query = www.google.com*” specifies the url to which you seek backlinks. With above query commands (4.1) and (4.2) to Google and Yahoo!, we can complete the legitimacy check.

In relevancy check, Trident first makes queries to the SEO tool server (*www.webmaster-toolkit.com*) for analyzing keywords of a page. For example, to analyze the keywords of *www.google.com*, a query command is composed to SEO keyword analyzer as:

$$/keyword - analysis - tool.shtml?url = http : //www.google.com. \quad (4.3)$$

In the query results, Trident obtains the most frequent keywords (with frequency higher than 3%) “google” and “search”, and uses these keywords in another *GET* request to get the web category in Yahoo! web directory (*dir.yahoo.com*). The query command is given as

$$/search/dir?p = google + search. \quad (4.4)$$

Note that keywords are specified in the command by “*p=google+search*”. The 1st returned result has a category path as: *Business and Economy > Business to Business > Communications and Networking > Internet and World Wide Web > Search and Navigation*. The top category is “Business and Economy”, which is also the category of *www.google.com*. With above http *GET* requests (4.3) and (4.4), Trident client completes the relevancy checks on the categories of the test page and its host site.

4.3 Performance Evaluation

In experiments of evaluating Trident, we have three data sets that contain urls of: actual phishing sites, legitimate sites, and legitimate foreign sites. Results and analysis are given below.

4.3.1 Experiment Data Sets

In addition to a phishing filter, we implement Trident as a Win32 application that can automatically evaluate urls in a data set. Three data sets are selected in our experiments:

- Phishing sites: PhishTank [39] was launched in October 2006 as an offshoot of OpenDNS. The company offers a community based phish verification system where users submit suspected phishes and other users “vote” if it is a phish or not. PhishTank has hundreds of phishing pages reported everyday, and some online phishing pages are verified as “valid phish” by users’ votes. We collected 387 phishes that are online and valid in September 2007. Those invalid or offline phishing pages were excluded.

- Legitimate web sites: Stanford WebBase project [46] has a crawler for grabbing web pages from a list of web sites. We manually verify 1223 legitimate sites from the list they used in August 2007.
- Foreign web sites: We got a list of UK web sites released by WEBSPAM-UK2006 [57]. In this list, 599 UK sites have been manually verified by two researchers and labeled as non-spam (legitimate). We use these foreign sites in our third data set.

We have four adjustable parameters: max search depth d , search breadth r , and PageRank threshold PR_{th} used in legitimacy check; and the number of categories used in relevancy check, N .

4.3.2 Detection against Phishing Pages

In the 1st experiment, we applied Trident on 387 valid phishes with fixed parameters $d = 3$ and $r = 3$, and adjustable parameters PR_{th} and N . We measured the miss detection rate, and the results are shown in Fig. 4.4. If we set a low threshold $PR_{th} = 3$ or 4, many phishing sites pass the legitimacy check, and the miss detection rate is high. If we set a high threshold $PR_{th} = 5$ in the legitimacy check, the miss detection rate could be reduced as low as 1% with 14 top categories used in relevancy check. In this case, four phishes passed the Trident legitimacy check because their host sites have PageRank scores higher than the threshold.

By inspecting the urls of these escaped phishes, we found that these phishers can compromise legitimate sites with very high PageRank scores and create pages on those sites. For example, one escaped phishing page is created on a legitimate host site.

[http : //site.voila.fr/pulix2/item9722396496.html](http://site.voila.fr/pulix2/item9722396496.html).

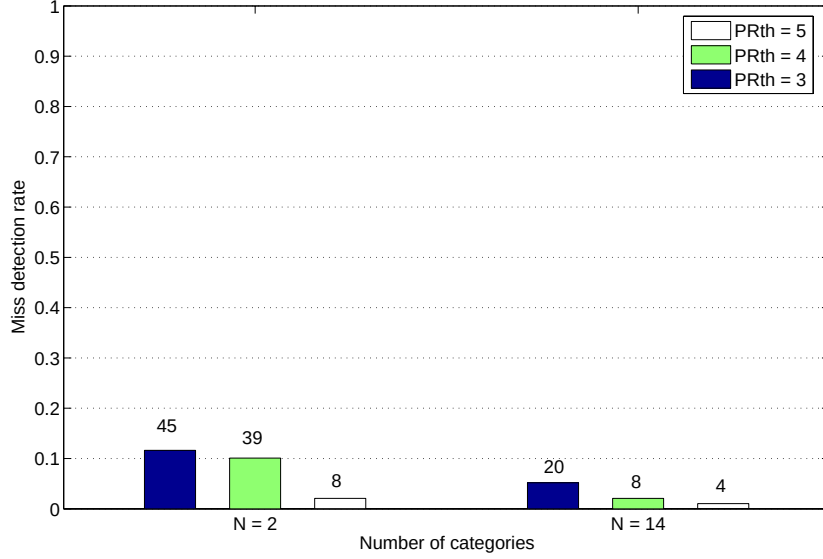


Figure 4.4: Miss detection rate of Trident when detecting phishing pages.

Although this phish is already dead, its host site “site.voila.fr” has a PageRank score of 6. Trident will determine it as legitimate and inaccessible, and let it pass through the legitimacy check. In fact, we found this site belongs to a web category different from the category of the phishing page that imitates a bank login form. This exceptional case can be detected by relevancy check if we apply relevancy check even if its host site has a PageRank score higher than the threshold. In this way, these phishes escaped in legitimacy check can be identified by the relevancy check.

We analyzed the experiment trace file under the setting: $d = 3$, $r = 3$, and $PRth = 5$. In Table 4.1, we show the values of search depth (*Depth*) when a phish is detected by Trident.

In Table 4.1, most of phishes are detected at $Depth = 1$, because most of phishes has no backlinks and fail the legitimacy check. Even if we select a small value (1 or 2) for d ,

Table 4.1: Search Depth when a phish is detected by Trident.

	$N = 2$	$N = 14$
$Depth = 0$	0	0
$Depth = 1$	373	375
$Depth = 2$	3	5
$Depth = 3$	3	3

Table 4.2: Number of phishes detected by Trident modules.

	$N = 2$	$N = 14$
Legitimacy and input form checks	372	372
Relevancy and input form checks	7	11

Trident can detect most phishes. However, some phishing pages may break into accessible legitimate sites. In Table 4.2, we show the number of phishes detected by different Trident modules. Trident’s relevancy check identified 7 and 11 phishes using different category classification methods. These pages have irrelevant topics to that of legitimate host sites.

Note that relevancy check does have miss detection if phishers create phishing pages with the same top category as an accessible legitimate host site. In this case, relevancy check need to further explorer the subcategory of this top category. If their subcategories are not matching, the page fails the relevancy check at a finer granularity. Due to the huge number of subcategories in each top category, the inspection of subcategories is not implemented in the current version of Trident.

4.3.3 Validations of Legitimate Sites

Since most legitimate sites on Internet are less known or less popular, it means most of them have low PageRank scores. Trident legitimacy check can distinguish these less

Table 4.3: Search Depth when a legitimate site is verified by Trident.

	$PRth = 5$	$PRth = 4$	$PRth = 3$
$Depth = 0$	1132	1158	1160
$Depth = 1$	86	63	62
$Depth = 2$	2	0	0
$Depth = 3$	0	0	0

Table 4.4: Proportion of sites verified by legitimate backlinks.

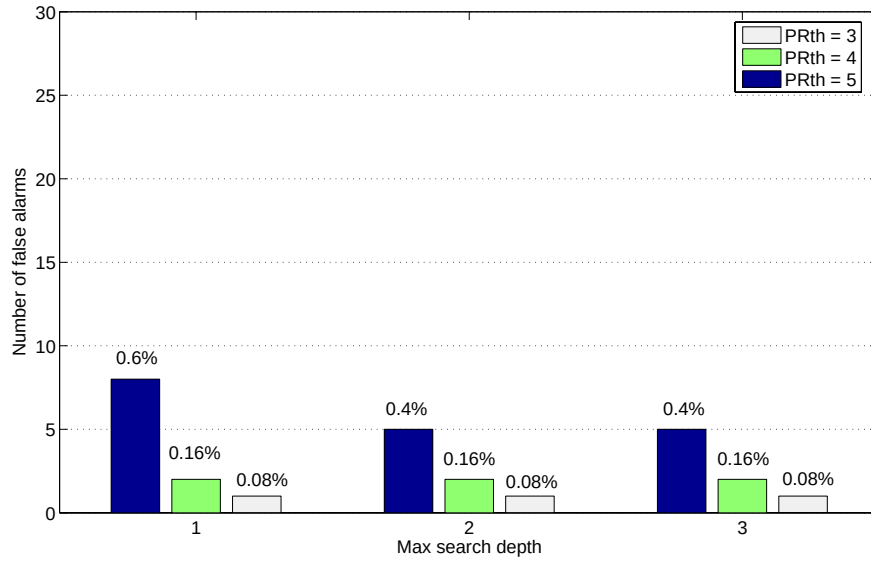
$PRth = 5$	$PRth = 4$	$PRth = 3$
7.2%	5.2%	5.1%

known but legitimate web sites from phishing sites by inspecting their backlinks.

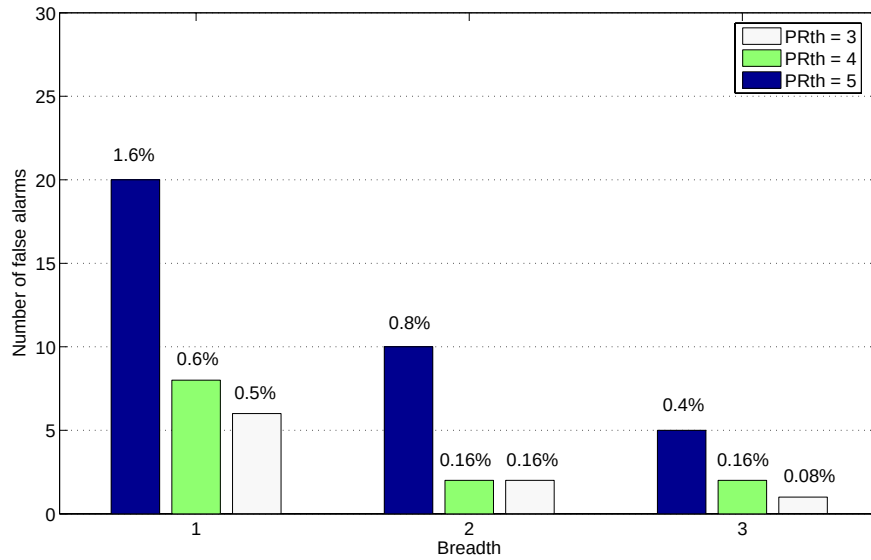
4.3.3.1 Legitimate Sites

WebBase crawler collects web pages from a list of more than 40,000 web sites every month. We select 1223 legitimate sites in their list for this experiment. We measure the false alarm rate (false positive). We adjusted three parameters: d , r , and $PRth$. The results are shown in Fig. 4.5, and we found the false alarm rates are low in all cases.

Under one experiment setting: $d = 3$, $r = 3$, and $PRth = 5$, we analyzed the experiment trace file. In Table 4.3, we show the values of search depth when a legitimate site is justified by Trident. Most sites are validated by legitimacy check at $Depth = 0$ since they have PageRank scores higher than the threshold. Other less known sites with low PageRank scores are validated by their legitimate backlinks. In Table 4.4, we found only a small portion of sites are verified with help of their legitimate backlinks.



(a) False positives, varying d and $PRth$;



(b) False positives, varying breadth r and $PRth$.

Figure 4.5: False alarms when evaluating legitimate web sites.

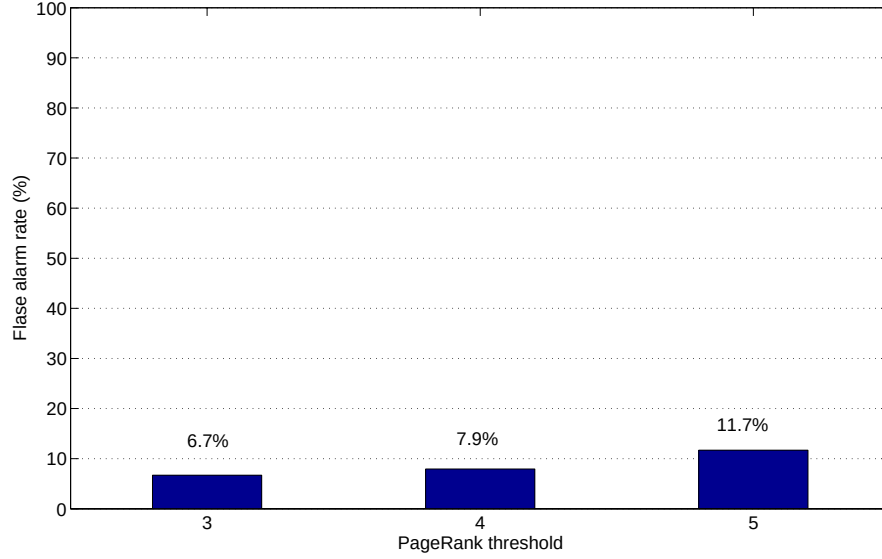


Figure 4.6: False alarm rate when evaluating legitimate UK sites by varying PR_{th} .

4.3.3.2 Legitimate Foreign Sites

WEBSHAM-UK2006 released a UK web site list, in which 599 sites are manually verified and tagged as non-spam. We applied Trident on these sites, and the metric is false alarm rate. We fixed both d and r as 3, and adjusted PR_{th} . In Fig. 4.6, the false alarm rate is higher than that in Figure. 4.5, since UK web sites are generally less known web sites for major search engines in US.

We also analyzed the experiment trace file. In Table 4.5, we show the values of search depth when a legitimate UK site is validated by Trident.

Compared with results in Table 4.3, we found that more web sites are validated as legitimate by the backlinks. It implies that validations of less known foreign sites rely on their legitimate backlinks. Concerning the false positives, the appropriate value for d should be at least 3 since many UK sites rely on the 3rd level backlinks. The proportion

Table 4.5: Search Depth when a legitimate UK site is verified by Trident.

	$PRth = 5$	$PRth = 4$	$PRth = 3$
$Depth = 0$	234	322	402
$Depth = 1$	175	149	104
$Depth = 2$	37	28	11
$Depth = 3$	10	3	0

Table 4.6: Proportion of UK sites verified by legitimate backlinks.

$PRth = 5$	$PRth = 4$	$PRth = 3$
48.7%	35.9%	22.2%

of UK sites verified by legitimate backlinks is summarized in Table 4.6. When $PRth$ is 5, almost half of foreign sites are justified by its legitimate backlinks.

Based on above results and analysis, we found that Trident is highly dependent on the legitimate backlinks, which is a more important factor than verifying the page content.

4.3.4 Comparisons with Other Anti-Phishing Tools

In the testing experiments of FireFox phishing protection [23], the researchers compare FireFox phishing protection tool [22] with Microsoft Phishing filter [30] on the same data set. Their data set includes 1040 phishing pages that are obtained from PhishTank. FireFox tool has a detection rate around 80% (i.e., a miss detection rate around 20%), and Microsoft phishing filter has a detection rate around 66%. We cannot use the same data set in our experiments because all phishing pages they used are already dead. In an alternative way, we collect online and valid phishes from the same source (PhishTank). In Fig. 4.4, we got a high detection rate 98-99%.

iTrustPage [41] is a FireFox add-on for phishing filtering. It checks a test page against a predefined white-list. If the page is not verified in the list and it contains an input form, iTrustPage will redirect users to a warning page asking the user to describe the form as if searching for the form on Google. Once the user enters keywords describing the form, iTrustPage performs a search on Google. If the form's top-level domain is found among the top 10 results, iTrustPage allows the user to pass through. This tool will obviously cause false positives since most pages are not contained in the white-list. We manually collect top 50 bank home pages from Yahoo! directory [51], and test these pages with iTrustPage v3.04. At the first access to these pages, iTrustPage redirects 39 bank home pages to the warning page with Google search help. The false positive rate is around 78%. Trident can readily validate all these bank pages with help of their legitimate backlinks. Although these are not absolutely fair comparisons, we are confident about the performance of Trident in terms of false negatives and false positives.

4.4 Discussions

In this section, we discussed several problems that may influence the reliability of Trident.

4.4.1 Manipulation of PageRank

The legitimacy check of Trident is dependent on the PageRank threshold PR_{th} . Is that possible for phishers to manipulate the PageRank algorithm and increase the phishing site's score higher than PR_{th} ? The answer is rarely possible. As known, link manipulation techniques can increase a web site's PageRank score higher than it deserves. Links may be manipulated by phishers in two ways: link spam and purchasing high quality links. By

link spam (or link farm), phishers can create a large number of boosting pages that are of low quality, and all these boosting pages link to a target web site. However, Google has updated their algorithm to defend against link spam. Google PageRank score is calculated continuously but is updated and released every two or three months. Even if link spam is successful in misleading the PageRank algorithm, phishing web sites are always dead before they get their PageRank score boosted, because most phishes are short-lived for only several hours or days. By purchasing high quality links from expired domains or some web masters, phishers have chances of boosting their PageRank scores. However, Google has announced that they will punish such link purchasing behaviors. Another challenge faced by phishers is that they need to know the update date of Google PageRank scores, and create their phishing sites linked by high quality links right before the update.

As discussed before, the tactic of phishers is to plant phishing pages on legitimate but accessible web sites, or break into legitimate web sites. These phishing pages usually have irrelevant topics to that of the host site, which can be identified by the relevancy check of Trident.

4.4.2 Problems in Relevancy Check

Currently, Trident only works for English written pages, since the SEO keyword extraction tool and Yahoo! directory are only compatible with English written pages. We need to find ways of obtaining the categories of pages written in other languages. For example, we can use web directory of a Chinese search engine for the web category of a Chinese written web page.

The accuracy of keyword extraction tool is another problem faced by relevancy check.

Of course, SEO tool is not the best keyword extractor since it may return useless keywords sometimes and mislead the Yahoo! directory to give inappropriate web categories. An accurate keyword extractor is necessary to select keywords of high representativeness of the web page rather than high frequency.

Chapter 5

POCKET: Parental Online Consent on Kids' Electronic Transactions

5.1 The POCKET Framework

The POCKET framework utilizes and extends the P3P framework by (1) incorporating a trusted third party (TTP) during interaction, (2) extending the merchant policy to include data elements as required by COPPA, the use and handling of data collected, and (3) automated exchange of personal information between the client and server. POCKET user agent allows merchants to identify the client as a child and automatically obtain parental authorization for information collected from the child.

Mont and Bromhall [33] and Mont et. al. [34] proposed increased user control over the disclosure of information and merchant accountability for using the data collected. Our technical solution extends this work by employing a stronger mechanism for ensuring merchant accountability. We employ a four entity architecture for security and enforcement.

The diagram shows an XML file structure for POCKET preferences. Annotations with arrows point to specific parts of the XML:

- User Preference**: Points to the `<PPF name="mike">` tag.
- Type of Information**: Points to the `name` attribute in `<PREF name="Full Name" option="yes">`.
- Option allowing or disallowing collection**: Points to the `option="yes"` attribute in the same `<PREF>` tag.
- Actual Data**: Points to the text `Mikey` inside the `<ITEM name="First Name">` tag.

```

<?xml version="1.0"?>
<POLICY name="POCKET v1.0 Alpha">
  <PPF name="mike">
    <PREF name="Full Name" option="yes">
      <ITEM name="First Name">Mikey</ITEM>
      <ITEM name="Last Name">Fierra</ITEM>
    </PREF>
    <PREF name="Only First Name" option="no"></PREF>
    <PREF name="Only Last Name" option="no"></PREF>
    . . . .
    <PREF name="Family Income" option="no"></PREF>
    <PREF name="Health Info" option="no"></PREF>
    <PREF name="Investment Info" option="no"></PREF>
  </PPF>
</POLICY>

```

Figure 5.1: Example POCKET preferences XML File.

The first two entities are the parent/guardian and the child. The parent/guardian (denoted as the “user”) enforces the privacy preference for the child (denoted as the “client”) and in this way provides a form of verifiable parental consent as required by COPPA. The privacy preference implemented on the client’s side is called the user privacy preference and the XML format file containing the preferences is called the user privacy preferences file (UPPF). The third entity is any merchant or web site the client visits. The privacy policies of the merchant is specified in a merchant privacy policy and recorded in a pre-specified location as the merchant privacy policy file (MPPF). The trusted third party (TTP) forms the fourth entity involved in POCKET.

The use of a TTP is derived from the 3-entity architecture proposed for identifier based encryption (IBE) [33]. In POCKET the role of a TTP is to provide mutual authentication—all the merchant and client to authenticate each other’s policies (similar to trust seals), public key distribution and assignment of a (one time) TTP signed ticket to each client

for access to POCKET compliant merchant web sites. TTP also serves as an enforcement agency in case of disputes between the merchant and the client.

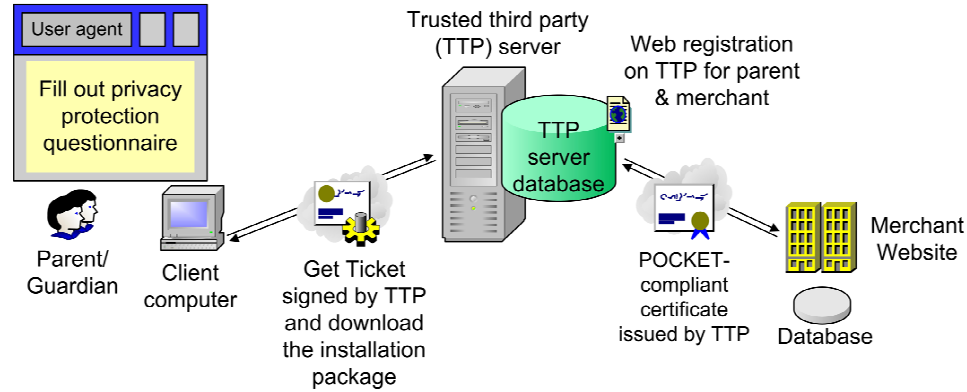


Figure 5.2: The POCKET registration and setup phase.

We propose the extension of P3P specification by including additional tags required for compliance with COPPA. All the key elements of the P3P vocabulary will remain the same. The DATA-GROUP and DATA tags will include additional categories to recognize the data collected from children. In addition to the users' financial information, it will include financial information of parents. New categories will also include information regarding age-group, sibling information, school or education institution, etc. In the current prototype, we use a simple yet effective XML-based vocabulary set instead of P3P. Figure 5.1 shows an example POCKET preferences file created at the client side. The merchant side policy includes only the PREF tag for each category of information and the corresponding option value.

Web sites provide hyper-text markup language (HTML) forms to collect the client's information. POCKET implements automatic information transfer and avoids forms completely. Some advantages of doing this in the context of protecting child privacy are: (1) It

prevents the web sites from collecting more information than specified in the policy. Forms may collect information that is optional and a child is not mature enough to know what information to disclose and what not to disclose. (2) The POCKET client (once validated) is guaranteed to provide only information that is absolutely necessary. (3) The information package is transmitted securely to the web site without the risk of being eavesdropped.

On the merchant's side, POCKET is implemented by augmenting web sites with an additional policy targeted towards children. This requires changes to the P3P policy file oriented towards the children visiting the web site and the information collected from them.

5.2 Implementation of POCKET

The software implementation of POCKET consists of a user agent (UA) and browser helper object (BHO). Once deployed on the client machine, the four entities exchange messages based on a protocol divided into three phases – *registration*, *setup* and *transaction*. The software installer package is available for download from the TTP server during the parent registration phase. The UA has two modes – parent-mode and child-mode. The software can be setup on the client machine in the parent-mode. During setup, the UA provides a questionnaire to the parent. The parent's responses are converted to an UPPF and stored on the client machine. When the UA enters the child-mode it installs the BHO and enforces the preferences. The BHO coupled with the browser can intercept user communication with the Internet. For security purposes, the UA requires the parent's password for any modification of the privacy preferences. The details of the phases in the POCKET framework is provided below.

5.2.1 Registration Phase

Figure 5.2 shows the pictorial representation of the registration phase. The user performs a one time registration accessing the TTP through a web site. During this phase the user registers with the TTP server and creates an account. Although not particularly specified in the POCKET protocol, parental verification can be implemented at the time of registration. The one time verification is possible because at account creation the parent creates a password that will be used for authentication at subsequent interactions. The TTP server assigns a unique ID to the user (parent/guardian). Registration provides the parent with the POCKET installer containing the UA, BHO and a TTP-signed ticket (used during the transaction phase).

The merchant also registers with the TTP server for participating in the POCKET protocol. The merchant provides answers to a questionnaire regarding the web sites' data collection practices and use policies. The answers to the questionnaire are converted into a machine readable XML format file and the MPPF is created. The merchant is required to deploy this file in a prescribed location on the merchant web site. During the transaction phase, this file is compared with the UPPF. The TTP also provides the merchant with a POCKET complaint certificate during registration. The client uses this certificate to authenticate the merchant during transaction phase.

5.2.2 Setup Phase

The user configures the POCKET UA by executing an installer and activates it before deploying it. The UA also requires the user to create a login and password (on the client machine). This login is only for the purpose of protecting the POCKET's configuration on

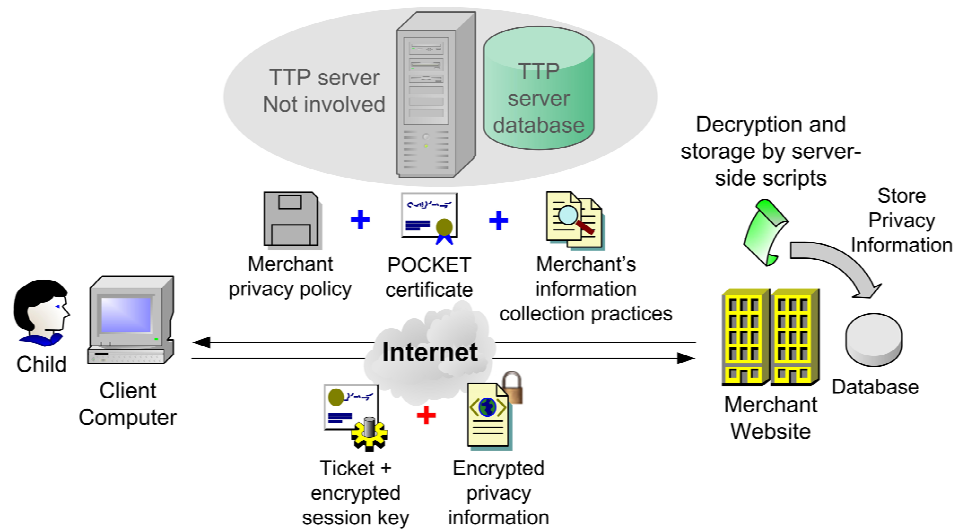


Figure 5.3: The POCKET transaction phase.

the client machine. The parent chooses the child's personal information that the merchant can collect. The user agent provides another dialog for the parent to enter the information. The client's information may include personal information (for example, full name, address and phone number), sibling data, school information among others. The parent can also configure the POCKET UA such that no information is collected from the child. The UA then converts the user's preference into a UPPF and stores it on the client's machine. The POCKET UA automatically enables the BHO after this configuration is complete. The BHO enforces the preferences specified in an UPPF.

5.2.3 Transaction Phase

Figure 5.3 shows the interaction between the client and the merchant web site during the transaction phase. When the client enters the merchant web site's uniform resource locator (URL) in the browser, the BHO installed on the client's browser (without client

involvement) requests the MPPF, the POCKET certificate and the merchant information collection practices. Here, we are assuming that the merchant complies with POCKET's requirement and places the MPPF in a pre-specified location. The POCKET agent decides to allow or block a web site after comparing the MPPF and UPPF. If the policy and preferences do not match, the BHO displays a "privacy policies do not match" message to the client and blocks the web site. If the MPPF and UPPF match, the client creates a *merchant specific* privacy information package (PIP). The PIP only includes the personal information requested by the merchant and is a subset of the information that was given consent (by the parent) to be disclosed. Using the proposed secure handshake protocol, the client side BHO uploads the PIP to the merchant site and allows the browser to display the web site. The POCKET agent on the client machine creates a *log file* entry showing the access. The release version of the POCKET software will include uploading of a digital contract¹ along with the PIP. The log file and the digital contract are useful in enforcing merchant accountability.

5.3 Security Features of the POCKET Framework

The data exchange protocol, especially the transaction phase, between the client and the merchant web site is vulnerable to several attacks. In this section, we analyze them and propose relevant countermeasures in the various phases of the client interaction with POCKET. Specifically, the transaction phase is vulnerable to unauthorized uploads and man-in-the-middle attacks.

¹ A digital contract is a legal agreement between the client and the merchant regarding the collection and use of information gathered.

Table 5.1: Legend of Terms Used

UID	–	a unique identification number (created for each user by TTP)
C_{pub}	–	client’s public key
C_{priv}	–	client’s private key
M_{pub}	–	merchant’s public key
TTP_{pub}	–	the public key of the TTP
$[message] SigTTP$	–	“message” signed by the TTP
Tik	–	a certified ticket (for authorized information exchange during transaction)
$[C_{pub}] SigTTP$	–	client’s public key signed by the TTP
TS	–	time stamp of the packet (verified to prevent replay attacks)
SK	–	session key for every client session
PIP	–	privacy information package contains the client’s personal information as required by a particular merchant
$E_k[message]$	–	“message” is encrypted using a key “ k ” (for example, $E_{M_{pub}}[SK]$ implies that SK is encrypted using M_{pub})
$\langle \dots \rangle$	–	denotes an entire message (for example, $\langle [C_{pub}] SigTTP \rangle$, denotes a message with the TTP-signed C_{pub})
$Tik TS \dots$	–	Tik appended with a TS

With POCKET system in place merchant web sites accept client’s privacy information package and store it for further processing. It is feasible for an anonymous user to upload spurious and harmful data to the merchant site, resulting in a Denial-of-Service (DoS) attack that prevents valid clients from accessing the web site. Without authenticating legitimate clients any data upload protocol fails to protect the merchant from attacks. As defined in the POCKET framework, the TTP implements a simplified Kerberos-like authentication mechanism [45] and provides mutual authentication between the client and the merchant. For mutual authentication, the TTP supplies a “certified ticket” for each client and a POCKET-compliant certificate for each merchant. The client’s certified ticket prevents malicious users from uploading harmful data to the merchant web site, while the POCKET-compliant certificate is verified (by the client) to determine merchant’s POCKET-compliance.

The man-in-the-middle attack is another type of attack that can be launched against

this system. This attack happens when a valid packet is intercepted and manipulated or processed for information. Potential problems with a man-in-the-middle attack are: (1) eavesdropping on sensitive and personal information, (2) information modification, and (3) packet replay at a later time. In addition to mutual authentication, the TTP performs the role of a key distributor and employs a pretty good privacy (PGP) framework [45]. With public-key exchange between a client and a merchant, the confidentiality and integrity of data can be protected by encryption with a session key and a digital signature respectively. The replay packets can be identified and ignored by employing a typical challenge/response handshake protocol. An attacker cannot eavesdrop on, or modify the contents of a valid packet. The following sections provide more details on how the proposed mechanisms are implemented in the POCKET framework.

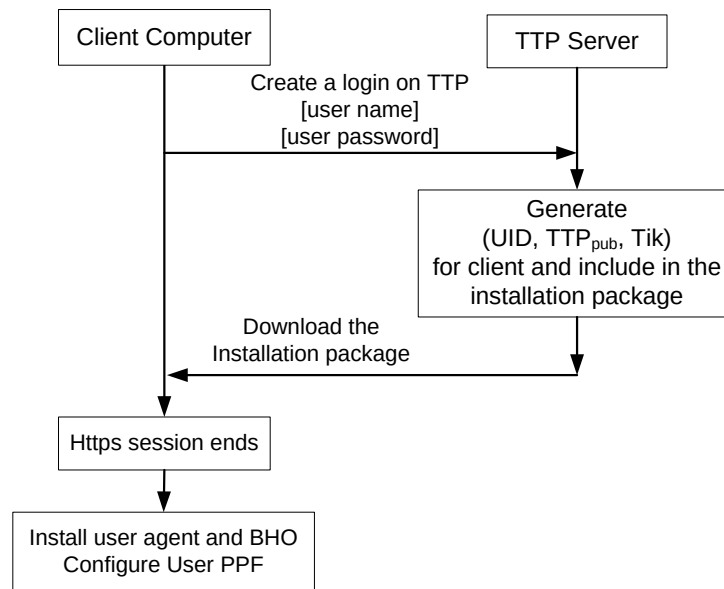


Figure 5.4: Building the POCKET installation package.

5.3.1 One Ticket Authentication Protocol

In POCKET, entity authentication between the client and the merchant is provided via an authentication protocol. The keys and tickets needed in the protocol are distributed by the TTP. Like the Kerberos [32, 47] authentication system, the POCKET's authentication protocol uses a ticket to authenticate the client to the merchant web site.

5.3.2 Registration Phase Ticket Distribution

Table 5.1 shows the legend of notations used in the following sections. In the registration phase, the user is required to create an account on the TTP server. This account creation is run inside a secure HTTP tunnel. The TTP server generates a public/private key pair (C_{pub} and C_{priv}), assigns a user ID (UID), and issues a certified service ticket (Tik). All this information is included in the installation package created for each authorized parent: UID, C_{pub} , C_{priv} , TTP_{pub} , ticket (Tik) and a signed public key, $[C_{pub}]SigTTP$. The parent downloads the installer before the end of the session. Figure 5.4 shows the installation package creation process on the TTP server.

The UA on the client's machine uses Tik to prove the client's identity. As we will see next, the public/private key pair solves the problem of unauthorized access to the merchant server.

5.3.3 Secure Transaction Phase

This section describes the protocol used for secure data exchange between the client and any POCKET-compliant merchant during the transaction phase. The protocol employs

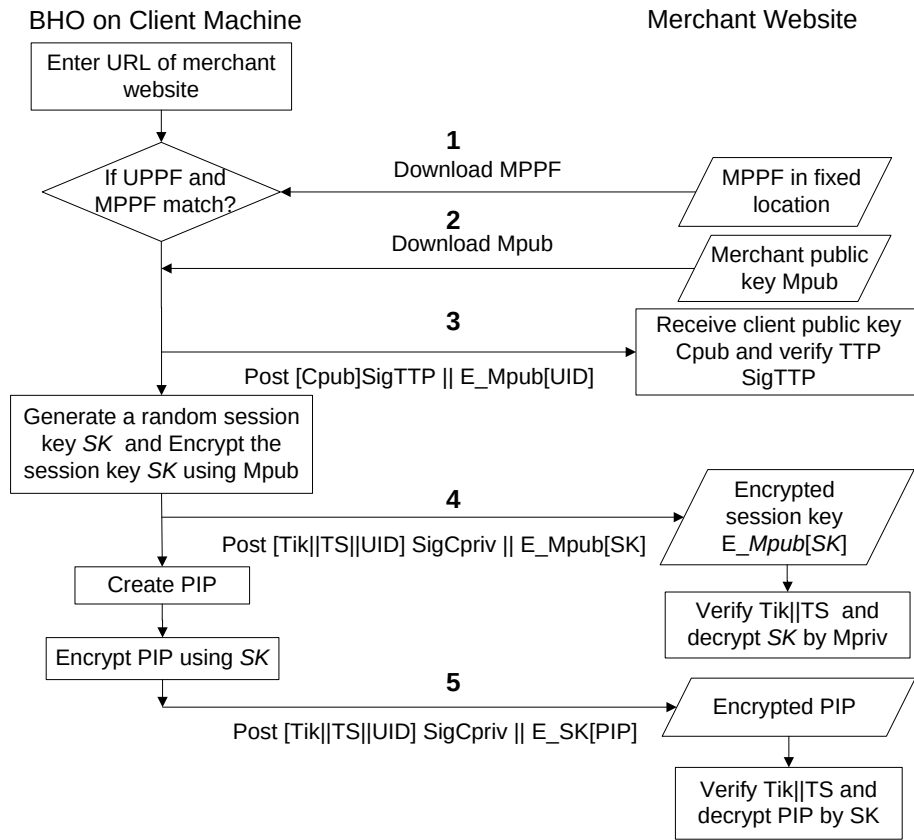


Figure 5.5: Transaction phase handshake protocol.

the pretty good privacy (PGP) [45] framework to provide *confidentiality*, *authenticity* and *integrity*. Specifically, if the merchant site is deemed POCKET compliant, a client machine uploads the PIP to the merchant. The confidentiality during this phase is provided by encrypting messages to be transmitted. Firstly, the client machine and the merchant exchange their public keys C_{pub} and M_{pub} , respectively. Secondly, the client machine generates a session key (SK) and uploads an $\langle E_{M_{pub}}[SK] \rangle$ message to the merchant server. Thirdly, the client encrypts the PIP with SK and sends it to the merchant. The merchant decrypts the PIP using the previously communicated SK . A digitally signed C_{pub} by the TTP ($[C_{pub}] SigTTP$) provides authenticity to a legitimate client. Integrity and non-repudiation of messages sent by the client to the merchant server are ensured by attaching a digital signature ($SigC_{priv}$) to each message. Only clients with a valid TTP-issued ticket (Tik) are allowed to upload data to the merchant server.

Figure 5.5 shows the actual messages exchanged in the handshake protocol of the transaction phase. The TTP is not involved during this handshaking. There are several potential security vulnerabilities in the public key exchange and PIP upload phases (steps 3, 4 and 5) shown in Figure 5.5. The merchant is subject to unauthorized uploads if the client uploads the C_{pub} directly. Thus, we use $[C_{pub}] SigTTP$, provided in the installation package. The UID encrypted using M_{pub} is also uploaded in the same message. In step 4 of the protocol, the TTP-signed Tik appended with TS and UID is signed with C_{priv} ($[Tik||TS||UID] SigC_{priv}$). A M_{pub} encrypted SK is also sent along with this message. The merchant server verifies $SigC_{priv}$ (using C_{pub}) to prevent unauthorized uploads. TS is used to prevent replay attacks. The merchant processes the message only after the message is found authentic and obtains SK . In step 5 the client machine uploads the SK encrypted

Privacy Preference File Editor

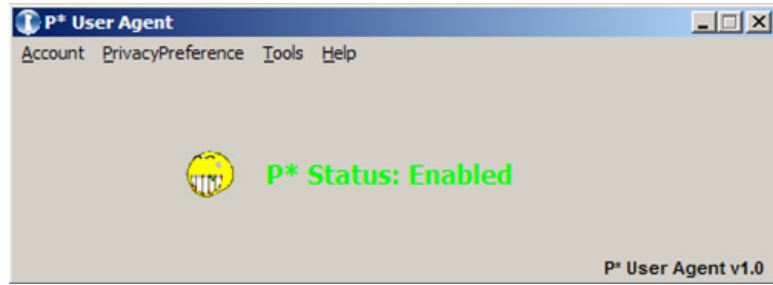
Child Name:

Parent or Guardian,
Please select from below to indicate which information types you authorize to be collected from your child by websites.

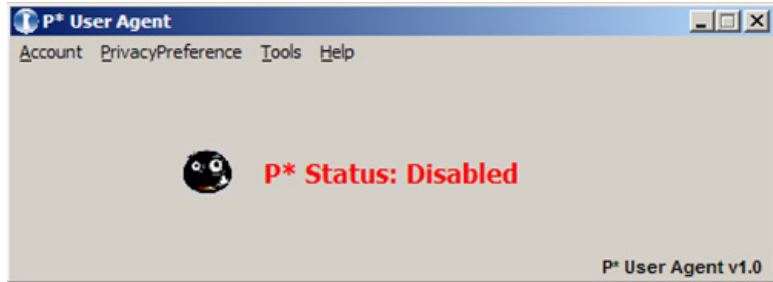
Preferences

☐ Full Name	☐ Parent occupation	☐ Child's email address
☒ Only first name	☐ Family income	☐ Parent's email address
☐ Only last name	☐ Investment information	☐ School attended
☐ Gender	☐ Parent's credit card number	☐ Instant messenger screen name or user ID
☐ Exact Age	☐ Mailing address	☐ Child's photograph
☒ Age range	☒ Zip code	☐ Child's video camera feed
☐ Age identifying questions	☐ Phone number	☐ Hobbies linked to personally identifiable information
☐ Birth Date	☐ Health information	☐ Homepage address
☐ Social security number	☐ Sibling's name	☐ Friend's name

Figure 5.6: Parent creating the UPPF for mike allowing only First Name, Age Range and Zip Code to be collected.



(a) Dialog of an enabled POCKET user agent.



(b) Dialog of a disabled POCKET user agent.

Figure 5.7: Status dialogs for the POCKET user agent.

PIP appended with $[Tik||TS||UID] SigC_{priv}$, to prevent unauthorized upload and replay attacks. The various security measures described above help make the handshake protocol of the transaction phase more robust against unauthorized uploads and man-in-the-middle attacks.

5.4 POCKET Software Prototype

We have completed a prototype of POCKET UA and BHO on Windows XP operating system. The current implementation of the BHO works for Microsoft's Internet Explorer version 6.0 (IE6) and can be easily extended for other versions and browsers. POCKET consists of a UA and BHO implemented in Visual C++. The UA is a simple dialog based

application that is used to configure POCKET on the client machine. The POCKET installer stores the contents C_{pub} , C_{priv} and $[C_{pub}] SigTTP$ at particular locations for the UA's use during the transaction phase. The parent configures POCKET with a setup password and protects the parent-mode from unauthorized access. After installation, the UA automatically presents the parent with the privacy preferences configuration dialog. By selecting the items the parent is allowing the merchant to collect particular information from the child. The UA converts the parent's selections into an UPPF and stores it in the client's machine. Figure 5.6 shows a screen shot of the UA taking input from the parent. In Figure 5.6 the parent allows collection of first name, age range and zip from the child and does not allow any of the other information to be collected. At this point the parent, by selecting none of the options, can implement the choice that NO information be collected from the child. The UA provides the parent with another dialog requesting the user to enter the actual information for the preference elements configured in the previous dialog (one shown in Figure 5.6). After configuration, the UA automatically enters the child-mode, starts the BHO and enforcing this UPPF when the client visits any web site. Figure 5.7(a) shows the dialog for the UA once setup and enabled on the client machine. Any shift from the child-mode to the parent-mode needs the parent's setup password. To prevent the client from disabling POCKET UA, closing UA also requires the parent's authentication. Figure 5.7(b)

A BHO is a DLL module designed as a plug-in to Microsoft's Internet Explorer web browser. The BHO API provides hooks to attach to the document object model (DOM) of the current page. The API controls the navigation within the page. Our BHO implements an XML parser for making comparisons between the MPPF and the UPPF. For XML

parsing we use the Xerces XML parser [50]. If the UPPF and the MPPF match, the BHO builds the PIP based on the merchant needs (specified in the MPPF) and uploads the PIP using a secure protocol to the merchant site. We use the Crypto++ API [12] to implement the security protocol of POCKET.

We have created mock merchant web sites that support the MPPF required by the POCKET BHO. We have tested the software implementation using these mocked web sites. We used a simplified XML file format that is consistent with the P3P specifications to create the privacy preference files.

Chapter 6

Conclusion

In this thesis, we present two new solutions for ensuring user online privacy: a client side solution to phishing attacks; and a four-entity framework for securing child's online privacy in e-transactions.

The client side anti-phishing solution is called Trident. It works as a browser plug-in that filters phishing web sites or phishing pages that break into legitimate sites. It does not require any local data to be stored, instead it makes queries on online reliable information repositories, which minimizes the chance of being compromised. Different from existing content analysis algorithms, the detection algorithm of Trident examine the legitimacy of backlinks to a web site and checks the web category relevancy between a web page and its host site. We applied Trident on three data sets for evaluating its performance. The results show that Trident outperforms existing anti-phishing tools by maintaining low rates of both false negatives and false positives. To the best of our knowledge, Trident is the first work of exploring backlinks and web categories for detection against phishes at the client side.

In addition, we propose a new privacy-enhancing framework called POCKET and have implemented a prototype for protecting children's privacy online. POCKET implements an automatic way to obtain verifiable parental consent as required by COPPA. It is an easy-to-use tool that technologically unsophisticated parents can deploy to protect their child's privacy. With POCKET user agent, parents can control the personal information collected by web sites from the child without constantly monitoring their activities online. The client's preferences and the merchant policy files have a format that is consistent with the P3P specifications. POCKET includes a secure protocol for uploading personal information from the client to the merchant. With mock merchant web sites, the prototype demonstrated its promise in offering a COPPA-compliant platform.

Bibliography

- [1] Bank of America, “SiteKey: Online Banking Security”, <http://www.bankofamerica.com/privacy/sitekey/>. Downloaded and archived on November 1, 2006.
- [2] BBBOnline (Privacy), <http://www.bbbonline.org/privacy/>.
- [3] P. Beatty, I. Reay, S. Dick and J. Miller, “P3P Adoption on E-Commerce Web sites: A Survey and Analysis”, *IEEE Internet Computing*, volume 11, number 2, 2007, pages 65–71.
- [4] F. Bélanger, J. S. Hiller and W. J. Smith, “Trustworthiness in electronic commerce: the role of privacy, security, and site attributes”, *Journal of Strategic Information Systems*, volume 11, number 3/4, December 2002, pages 245–270.
- [5] Children’s Advertising Review Unit, <http://www.caru.org/>.
- [6] N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell, “Client-side defense against web-based identity theft”, in *Proc. of the Network and Distributed System Security Symposium (NDSS ’04)*, February 2004.
- [7] CPA Webtrust, <http://www.webtrust.net/>.
- [8] L. F. Cranor, S. Byers and D. Kormann, “An Analysis of P3P Deployment on Commercial, Government, and Children’s Web Sites as of May 2003”, AT&T Labs-Research, Prepared for the Federal Trade Commission Workshop on Technologies for Protecting Personal Information, May 2003.
- [9] L. F. Cranor, “P3P: Making Privacy Policies More Useful”, *IEEE Security and Privacy*, volume 1, number 6, 2003, pages 50–55.
- [10] L. Cranor and et. al., “The Platform for Privacy Preferences 1.1 (P3P1.1) Specification”, Rigo Wenning and Matthias Schunter, November 2006. Available: <http://www.w3.org/TR/P3P11/>.

- [11] L. F. Cranor, P. Guduru and M. Arjula, "User Interfaces for Privacy Agents", *ACM Transactions on Computer-Human Interaction*, volume 13, number 2, 2006, pages 135–178.
- [12] "Crypto++ 5.4", Available for download at <http://www.cryptopp.com>.
- [13] R. Dhamija and J. D. Tygar, "The Battle Against Phishing: Dynamic Security Skins", in *Proceedings of the Symposium on Usable Privacy and Security (SOUPS '05)*, July 2005, pages 77–88.
- [14] Electronic Privacy Information Center and Junkbusters, "Pretty Poor Privacy: An Assessment of P3P and Internet Privacy", <http://www.epic.org/reports/prettypoorprivacy.html>, June 2000.
- [15] Electronic Privacy Information Center and Junkbusters, "Public Opinion on Privacy", April 2007, <http://www.epic.org/privacy/survey/default.html>.
- [16] A. Emigh, "Online Identity Theft: Phishing Technology, Chokepoints and Countermeasures", ITTC Report on Online Identity Theft Technology and Countermeasures. October 3, 2005.
- [17] Entertainment Software Rating Board, <http://www.esrb.org/index-js.jsp>.
- [18] D. Esposito, "Browser Helper Objects: The Browser the Way You Want It", Microsoft Corporation, January 1999, <http://msdn2.microsoft.com/en-us/library/bb250436.aspx>.
- [19] Federal Trade Commission, "FTC seeks comment on proposed COPPA rule amendment", January 2005, <http://www.ftc.gov/opa/2005/01/coppafrn.htm>.
- [20] Federal Trade Commission, "Privacy Online: A Report to Congress", <http://www.ftc.gov/reports/privacy3/toc.htm>, June 1998.
- [21] Federal Trade Commission, "Xanga.com to Pay \$1 Million for Violating Children's Online Privacy Protection Rule", <http://ftc.gov/opa/2006/09/xanga.htm>, September, 2006.
- [22] FireFox, "Phishing Protection", <http://www.mozilla.com/en-US/firefox/phishing-protection/>.
- [23] Firefox 2 Phishing Protection Effectiveness Testing, <http://www.mozilla.org/security/phishing-test.html>.

- [24] M. Jakobsson, and S. Stamm, “Invasive browser sniffing and countermeasures”, in *Proceedings of the 15th International Conference on World Wide Web (WWW '06)*, May, 2006, pages 523–532.
- [25] D. M. Kristol, “HTTP Cookies: Standards, privacy, and politics”, *ACM Trans. Inter. Tech.*, volume 1, number 2, 2001, pages 151–198.
- [26] S. Kuchinskas, “In Web Sites We Trust?”, *Brandweek*, February, 1999.
- [27] L. Carter and F. Bélanger, “Citizen Adoption of Electronic Government Initiatives”, *HICSS '04: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 5*, January, 2004.
- [28] Messaging Anti-Abuse Working Group (MAAWG), and Anti-Phishing Working Group (APWG), “Anti-Phishing Best Practices for ISPs and Mailbox Providers”, Version 1.01, July 2006.
- [29] Microsoft, “Internet Explorer 7.0 Features in Windows Vista: Parental Controls”, <http://www.microsoft.com/windows/products/windowsvista/features/details/parentalcontrols.aspx>.
- [30] Microsoft, “Microsoft Phishing Filter: A New Approach to Building Trust in E-Commerce Content”, white paper, 2005.
- [31] Microsoft, “Privacy in Internet Explorer 6”, <http://msdn.microsoft.com/workshop/security/privacy/overview/privacyIE6.asp>.
- [32] S. P. Miller, B. C. Neuman and J. I. Schiller and J. H. Saltzer, “Section E.2.1: Kerberos Authentication and Authorization System”, Massachusetts Institute of Technology, December 1987.
- [33] M. C. Mont and P. Bramhall, “IBE Applied to Privacy and Identity Management Trusted”, HP Laboratories, Bristol, May, 2003.
- [34] M. C. Mont, S. Pearson and P. Bramhall, “Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services”, HP Laboratories, Bristol, March 2003.
- [35] MSDN, “Http Sessions”, <http://msdn2.microsoft.com/en-US/library/aa384322.aspx>.
- [36] Net Nanny, “Net Nanny User Guide”, <http://www.netnanny.com/store/nav.asp?v=1.0&page=nn5-userguide>

- [37] “Net Users Distrust Corporate Privacy Policies-Study”, *Newsbytes*, 2002.
- [38] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank Citation Ranking: Bringing Order to the Web”, Stanford Digital Library Technologies Project, 1998.
- [39] PhishTank, <http://www.phishtank.com/>.
- [40] PRivacy Vaults Online, <http://www.privo.com/>.
- [41] T. Ronda and S. Saroiu, “iTrustPage: Pretty Good Phishing Protection”, Technical Report, University of Toronto, December 2006.
- [42] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer, “The Emperor’s New Security Indicators”, in Proceedings of *IEEE Symposium on Security and Privacy*, (SP ’07), May 2007, pages 20–23.
- [43] Search engine optimizers (SEO), Keyword analysis tool, <http://www.webmaster-toolkit.com/keyword-analysis-tool.shtml>.
- [44] C. V. Slyke, F. Bélanger and C. L. Comunale, “Adopting Business-to-Consumer Electronic Commerce: The Effects of Trust”, *Database for Advances in Information Systems*, volume 35, Spring, 2004.
- [45] W. Stallings, *Cryptography and Network Security, Principles and Practices*, Pearson Education Inc., Third Edition, 2003.
- [46] Stanford WebBase project, available at: <http://dbpubs.stanford.edu:8091/~testbed/doc2/WebBase/>.
- [47] J. G. Steiner, B. C. Neuman and J. I. Schiller, “Kerberos: An Authentication Service for Open Network Systems”, *Proceedings of the USENIX Winter 1988 Technical Conference*, USENIX Association, 1988 pages 191–202.
- [48] Studio Archetype/Sapient and Cheskin Research, “eCommerce Trust Study”, 1999.
- [49] H. Taylor, “Why Some Companies Are Trusted and Others Are Not: Personal Experience and Knowledge of Company More Important than Glitz”, June 2001, http://www.harrisinteractive.com/harris_poll/index.asp?PID=237.
- [50] The Apache XML Project, “Xerces-C++”, Available for download at <http://xml.apache.org/xerces-c/>.
- [51] Top 50 bank websites at Yahoo! search, <http://search.yahoo.com/search/dir?p=bank>.

- [52] TRUSTe, <http://www.truste.org/>
- [53] “TRUSTe”, “Children’s Privacy Seal Program Requirements”, <http://www.truste.org/requirements.php#req2>.
- [54] TrustGauge, <http://www.trustgauge.com/>.
- [55] J. Turow, L. Feldman and K. Meltzer, “Open to Exploitation: American Shoppers Online and Offline”, Annenberg Public Policy Center of the University of Pennsylvania, June 2005.
- [56] W3C, “Extensible Markup Language (XML) 1.1”, Tim Bray and Jean Paoli and et. al., Second Edition, September 2006, <http://www.w3.org/TR/xml11>.
- [57] WEBSPAM-UK2006, Human assigned labels for a set of hosts, <http://www.yr-bcn.es/webspam/datasets/>.
- [58] Wikipedia, “Spam (electronic)”, [http://en.wikipedia.org/wiki/Spam_\(electronic\)](http://en.wikipedia.org/wiki/Spam_(electronic)).
- [59] WRAAC.org, “ParentalControl Toolbar”, <http://www.parentalcontrolbar.org/index.htm>.
- [60] M. Wu, R. C. Miller and S. L. Garfinkel, “Do Security Toolbars Actually Prevent Phishing Attacks?”, in Proceedings of *Conference on Human Factors in Computing Systems (CHI '06)*, April 2006, pages 601–610.
- [61] M. Wu, R. C. Miller and G. Little, “Web Wallet: Preventing Phishing Attacks by Revealing User Intentions”, in Proceedings of *Symposium on Usable Privacy and Security (SOUPS '06)*, July 2006, pages 102–113.
- [62] Yahoo! Inc. Yahoo! “Personalized Sign-In Seal”, 2006, <https://protect.login.yahoo.com/>.
- [63] Yahoo! Site Explorer Inbound Links API, <http://developer.yahoo.com/search/siteexplorer/V1/inlinkData.html>.

Acknowledgements

I would like to express my deep-felt gratitude to my advisor, Dr. Jung-Min Park, for his advice, encouragement, enduring patience and constant support. He always provides clear explanations when I was hopelessly lost, and constantly drives me with energy when I was tired.

I also wish to thank the other members of my committee, Dr. Y. Thomas Hou and Dr. Yaling Yang, both at Virginia Tech. Their suggestions, comments and additional guidance were invaluable to the completion of this work.