Scientific Research

HumanBoost: Utilization of Users' Past Trust Decision for Identifying Fraudulent Websites

Daisuke Miyamoto[1], Hiroaki Hazeyama[2], Youki Kadobayashi[2]

[1]Information Security Research Center, National Institute of Information and Communications Technology, Koganei, Tokyo, Japan;
[2]Graduate School of Information Science, Nara Advanced Institute of Science and Technology, Ikoma, Nara, Japan.
E-mail: daisu-mi@nict.go.jp, hiroa-ha@is.naist.jp, youki-k@is.aist-nara.ac.jp

ABSTRACT

This paper presents HumanBoost, an approach that aims at improving the accuracy of detecting so-called phishing sites by utilizing users' past trust decisions (PTDs). Web users are generally required to make trust decisions whenever their personal information is requested by a website. We assume that a database of user PTDs would be transformed into a binary vector, representing phishing or not-phishing, and the binary vector can be used for detecting phishing sites, similar to the existing heuristics. For our pilot study, in November 2007, we invited 10 participants and performed a subject experiment. The participants browsed 14 simulated phishing sites and six legitimate sites, and judged whether or not the site appeared to be a phishing site. We utilize participants' trust decisions as a new heuristic and we let AdaBoost incorporate it into eight existing heuristics. The results show that the average error rate for HumanBoost was 13.4%, whereas for participants it was 19.0% and for AdaBoost 20.0%. We also conducted two follow-up studies in March 2010 and July 2010, observed that the average error rate for HumanBoost was below the others. We therefore conclude that PTDs are available as new heuristics, and HumanBoost has the potential to improve detection accuracy for Web user.

Keywords: *Phishing, Personalization, AdaBoost, Trust Decision*

1. Introduction

Phishing is a form of identity theft in which the targets are users rather than computer systems. A phishing attacker attracts victims to a spoofed website, a so-called phishing site, and attempts to persuade them to provide their personal information. Damage suffered from phishing is increasing. In 2005, the Gartner Survey reported that 1.2 million consumers lost $929 million as a result of phishing attacks [1]. The modern survey conducted in 2008 also reported that more than 5 million consumers lost $1.76 billion [2]. The number of phishing sites is also increasing. According to trend reports published by the Anti-Phishing Working Group [3], the number of the reported phishing sites was 25,630 in March 2008, far surpassing the 14,315 in July 2005.

To deal with phishing attacks, a heuristics-based detection method has begun to garner attention. A heuristic is an algorithm to identify phishing sites based on users' experience, and checks whether a site appears to be a phishing site. Checking the life time duration of the issued website is well-known heuristic as most phishing

sites' URL expired in short time span. Based on the detection result from each heuristic, the heuristic-based solution calculates the likelihood of a site being a phishing site and compares the likelihood with the defined discrimination threshold. Unfortunately, the detection accuracy of existing heuristic-based solutions is nowhere near suitable for practical use [4] even though there exists various heuristics discovered by former studies. In our previous work [5], we attempted to improve this accuracy by employing machine learning techniques for combining heuristics, since we assumed that the inaccuracy is caused by heuristics-based solutions that cannot use the heuristics appropriately. In most cases, machine learning-based detection methods (MLBDMs) performed better than existing detection methods. Especially, an AdaBoost-based detection method showed the highest detection accuracy.

In this paper, we propose HumanBoost, which aims at improving AdaBoost-based detection methods. The key concept of HumanBoost is utilizing Web users' past trust decisions (PTDs). Basically, humans have the potential to identify phishing sites, even if existing heuristics can-

not detect them. If we can construct a database of PTDs for each Web user, we can use the record of the user's trust decisions as a feature vector for detecting phishing sites. HumanBoost also involves the idea of adjusting the detection for each Web user. If a user is a security expert, the most predominant factor on detecting phishing sites would be his/her trust decisions. Conversely, the existing heuristic will have a strong effect on detection when the user is a novice and his/her PTD has often failed.

In our study in November 2007, we invited 10 participants and performed a subject experiment. The participants browsed 14 simulated phishing sites and six legitimate sites, and judged whether or not the site appeared to be a phishing site. By utilizing participants' trust decisions as a new weak-hypothesis, we let AdaBoost incorporate the heuristic into eight existing heuristics. The results show that the average error rate for HumanBoost was 13.4%, whereas that for participants was 19.0% and for the AdaBoost-based detection method 20.0%. We then conducted a follow-up study in March 2010. This study had 11 participants with the almost same conditions as the first. The results show that the average error rate for HumanBoost was 10.7%, whereas that for participants was 31.4% and for AdaBoost 12.0%. We also invited 309 participants and performed another follow-up study in July 2010. The results show that the average error rate for HumanBoost was 9.7%, whereas that for participants was 40.5% and for AdaBoost 10.5%.

The rest of this paper is organized as follows. Section 2 summarizes the related work, and Section 3 explains our proposal. Section 4 describes our preliminary evaluation, and Section 5 presents a follow-up study. Section 6 discusses the availability of PTDs, the way for removing bias, and issues on implementing HumanBoost-capable system. Finally, Section 7 concludes our contribution.
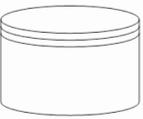
2. Related Work

For mitigating phishing attacks, machine learning, which facilitates the development of algorithms or techniques by enabling computer systems to learn, has begun to garner attention. PFILTER, which was proposed by Fette

et al. [6], employed SVM to distinguish phishing emails from other emails. Abu-Nimeh *et al.* compared the predictive accuracy of six machine learning methods [7]. They analyzed 1,117 phishing emails and 1,718 legitimate emails with 43 features for distinguishing phishing emails. Their research showed that the lowest error rate was 7.72% for Random Forests. Ram Basnet *et al.* performed an evaluation of six different machine learning-based detection methods [8]. They analyzed 973 phishing emails and 3,027 legitimate emails with 12 features, and showed that the lowest error rate was 2.01%. The experimental conditions were differed between them, but machine learning provided high accuracy for the detection of phishing emails.

Apart from phishing emails, machine learning was also used to detect phishing sites. Pan *et al.* presented an SVM-based page classifier for detecting those websites [9]. They analyzed 279 phishing sites and 100 legitimate sites with eight features, and the results showed the average error rate to be 16%. Our previous work employed nine machine learning techniques [5], AdaBoost, Bagging, Support Vector Machines, Classification and Regression Trees, Logistic Regression, Random Forests, Neural Networks, Naïve Bayes, and Bayesian Additive Regression Trees. We also employed eight heuristics presented in [10] and analyzed 3,000 URLs, consisting of 1,500 legitimate sites and the same number of phishing sites, reported on PhishTank.com [11] from November 2007 to February 2008. Our evaluation results showed the highest f_1 measure at 0.8581, lowest error rate at 14.15% and highest AUC at 0.9342; all of which were observed for the AdaBoost-based detection method. In most cases, MLBDMs performed better than the existing detection method.

Albeit earlier researches used machine learning, we find that there are two problems. One is the number of features for detecting phishing sites is less than that for detecting phishing emails. It indicates that the development of new heuristic for phishing sites is more difficult than that for phishing emails. The other is to lack the idea of protecting individual Web user. We considered that the protection methods should differ in each Web

	URL	Actual Condition	The user's trust decision	Heuristics #1	...	Heuristics #N
	Site 1	Phishing	Phishing	Phishing	...	Legitimate
	Site 2	Phishing	Legitimate	Phishing	...	Phishing
	Site 3	Phishing	Phishing	Legitimate	...	Phishing
PTD database	...	...	...	...	...	...
	Site M	Legitimate	Legitimate	Legitimate	...	Phishing

Figure 1. Example of PTD and its scheme.

user as long as phishing attacks target individual users. Our proposed HumanBoost aims at using past trust decisions as a new heuristic. It also enables the detection algorithm to customize for each Web user by machine learning processes describe in Section 3.2.

3. HumanBoost

3.1. Overview

The key concept of HumanBoost is utilizing Web users' past trust decisions (PTDs). Web users are generally required to make trust decisions whenever they input their personal information into websites. In other words, we assumed that a Web user outputs a binary variable, phishing or legitimate, when the website requires users to input their password. Note that existing heuristics for detecting phishing sites, which we explain in Section 4.2, are similar to output binary variables denoting phishing or not-phishing.

In HumanBoost, we assume that each Web user has his/her own PTD database, as shown in **Figure 1**. The schema of the PTD database consists of the website's URL, actual conditions, the result of the user's trust decision, and the results from existing heuristics. Note that we do not propose sharing the PTD database among users due to the privacy concerns. The PTD database can be regarded as a training dataset that consists of $N + 1$ binary explanatory variables and one binary response variable. We, therefore, employ a machine learning technique for studying this binary vector for each user's PTD database.

3.2. Theoretical Background

In this study we employ the Adaptive Boosting (AdaBoost) [12] algorithm that learns a strong algorithm by combining a set of weak algorithms h_t and a set of weight α_t:

$$H_{ada} = \sum h_t \times \alpha_t \tag{1}$$

The weights are learned through supervised training off-line. Formally, AdaBoost uses a set of input data $\{ x_i, y_i : i = 1, \ldots , m\}$ where x_i is the input and y_i is the classification.

Each weak algorithm is only required to make the correct detections in slightly over half the time. The AdaBoost algorithm iterates the calculation of a set of weight $D_t(i)$ on the samples. At $t = 1$, the samples are equally weighted so $D_t(i) = 1/m$.

The update rule consists of three stages. First, AdaBoost chooses the weight as shown in (2).

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right) \tag{2}$$

where $\varepsilon_t = Pr_{i \sim Di}[h_t(x_i) \neq y_i]$. Second, AdaBoost updates

the weights by (3).

$$D_{t+1} = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} \text{ if } h_t(x_i) = y_i \\ e^{\alpha_t} \text{ if } h_t(x_i) \neq y_i \end{cases} \tag{3}$$

where Z_t is a normalization, factor, $\sum D_{t+1}(i) = 1$. Finally, it outputs the final hypothesis as shown in (1).

We have two reasons of employing AdaBoost. One is that it had performed better in our previous comparative study [5], where it demonstrated the lowest error rate, the highest f_1 measure, and the highest AUC of the AdaBoost-based detection method, as mentioned in Section 2.

The other is that we expect AdaBoost to cover each user's weak points. Theoretically, the boosting algorithms assign high weight to a classifier that correctly labels a site that other classifiers had labeled incorrectly, as shown in (3). Assuming that a user's trust decision can be treated as a classifier, AdaBoost would cover users' weak points by assigning high weights to heuristics that can correctly judge a site that the user is likely to misjudge.

4. Experiment and Results

To check the availability of PTDs, we invited participants and performed a phishing IQ test to construct PTDs, in November 2007. This section describes the dataset description of the phishing IQ test, introduces the heuristics that we used, and then explains our experimental design and finally show the results.

4.1. Dataset Description

Similar to the typical phishing IQ tests performed by Dhamija *et al*. [13], we prepared 14 simulated phishing sites and six legitimate ones, all of which contained Web forms in which users could input their personal information such as user ID and password. The conditions of the sites are shown in **Table 1**.

Website 1, 4, 7, 12, 13, 19 were actual company websites, but these sites contained defective features that could mislead participants into labeling them as phishing. Websites 1 and 12 required users to input their password, though they employed no SSL certification. Website 4 was Goldman Sachs with the domain name *webid 2.gs.com*. Since "gs" can imply multiple meanings, the domain name can confuse participants. Similarly, website 13 contained "clientserv" in its domain name. Website 7 was Nanto Bank, a Japanese regional bank mainly operating in Nara Prefecture, where almost all the participants lived, but its domain name *www2.paweb. anser.or.jp* which gives no indication of the bank's name. Website 19 was Apple Computer Inc. and employed a valid SSL, but web browsers displayed an alert window because of its accessing non-SSL content.

The rests were phishing sites. Websites 5, 11, and 15 were derived from actual phishing sites based on a report from Phishtank.com. Other phishing sites were simulated phishing sites that mimic actual websites by using the phishing practices described in the followed sections.

4.1.1. Confusing URL

Websites 2, 6, 16, and 17 were made to look like well-known sites, but with slightly different or confusing URLs. A phishing attacker (phisher) registering a similar or otherwise legitimate-sounding domain name such as *www-bk-mufg.jp* is increasingly common. Website 6 was hosted at *www.bankofthevvest.com*, with two "v"s instead of a "w" in its domain name. According to the phishing IQ test conducted by Dhamija [13], the phishing site that fooled the most participants was an exact replica of the Bank of the West homepage and hosted at this domain name.

4.1.2. IP Address Abuse

Websites 10 and 14 employed IP address abuse; instead of showing the domain name, the IP address appears in the browsers' address bar. For website 10, a phisher copied the contents of the actual Citibank homepage into a website and created URLs using IP addresses. The IP address does not point Citibank, but some participants would not be aware of this and think the site is legitimate.

4.1.3. IDN Abuse

Websites 9 and 18 employed International Domain Name (IDN) abuse, modern phishing technique. Fu *et al*. indicated [14] that the letter "a" in the Cyrillic alphabet is instance, the URL of website 9 is *www.xn--pypal-4ve. com*, which is clearly different from *www.paypal.com*. Yet, the domain name can be shown as a *www.paypal. com* in web browsers.

Table 1. Conditions of each website.

#	Website	Real / Spoof	Lang	Description
1	Live.com	real	EN	URL (*login.live.com*)
2	Tokyo-Mitsubishi UFJ	spoof	JP	URL (*www-bk-mufg.jp*), similar to the legitimate URL (*www.bk.mufg.jp*)
3	PayPal	spoof	EN	URL (www.paypal.com.%73%69 ... %6f%6d) (URL Encoding Abuse)
4	Goldman Sachs	real	EN	URL (*webid2.gs.com*), SSL
5	Natwest Bank	spoof	EN	URL (*onlinesession-0815.natwest.com.esb6eyond.gz.cn*), derived from PhishTank.com
6	Bank of the West	spoof	EN	URL (*www.bankofthevvest.com*), similar to the legitimate URL (*www.bankofthewest.com*)
7	Nanto Bank	real	JP	URL (*www2.paweb.anser.or.jp*), SSL, third party URL
8	Bank of America	spoof	EN	URL (*bankofamerica.com@index.jsp-login-page.com*) (URL Scheme Abuse)
9	PayPal	spoof	EN	URL (*www.paypal.com*), first "a" letter is a Cyrillic small letter "a" (U+430) (IDN Abuse)
10	Citibank	spoof	EN	URL (IP address) (IP Address Abuse)
11	Amazon	spoof	EN	URL (*www.importen.se*), contains "amazon" in its path, derived from PhishTank.com
12	Xanga	real	EN	URL (*www.xanga.com*)
13	Morgan Stanley	real	EN	URL (*www.morganstanleyclientserv.com*), SSL
14	Yahoo	spoof	EN	URL (IP address) (IP Address Abuse)
15	U.S.D. of the Treasury	spoof	EN	URL (*www.tarekfayed.com*), derived from PhishTank.com
16	Sumitomo Mitsui Card	spoof	JP	URL (*www.smcb-card.com*), similar to the legitimate URL (*www.smbc-card.com*)
17	eBay	spoof	EN	URL (*secuirty.ebayonlineregist.com*)
18	Citibank	spoof	EN	URL (シティバンク.com), is pronounced "Shi Tee Ban Ku", look-alike "Citibank" in Japanese Letter) (IDN Abuse)
19	Apple	real	EN	URL (*connect.apple.com*), SSL, popup warning by accessing non-SSL content
20	PayPal	spoof	EN	URL (*www.paypal.com@verisign-registered.com*), (URL Scheme Abuse)

4.1.4. URL Scheme Abuse

The URLs of websites 8 and 20 contained an at-mark (@) symbol. When the symbol is used in a URL, all text before it is ignored and the browser references only the information following it as a hostname. For website 8, the URL is

http://bankofamerica.com@index.jsp-login-page.com.

Even if it seemed like *bankofamerica.com*, web browsers would ignore this and would be directed to *index.jsp-login-page.com*.

4.1.5. URL Encoding Abuse

URL encoding is an accepted method of representing characters within a URL that may need special syntax handling to be correctly interpreted. This is achieved by encoding the character to be interpreted with a sequence of three characters. This triplet sequence consists of the percent character "%" followed by the two hexadecimal digits representing the octet code of the original character. For instance, the US-ASCII character set represents a letter "s" with hexadecimal code 73, so its URL-encoded representation is %73. Website 3 glossed over its domain name by URL encoding abuse to make the domain name mimic that of PayPal, Inc.

4.2. Heuristics

Our experiment employs eight types of heuristics, all of which were employed by CANTINA [15]. To the best of our knowledge, CANTINA is the most successful tool for combining heuristics, since it has achieved high accuracy in detecting phishing sites without using the URL blacklist.

4.2.1. Age of Domain (H_1)

This is a check of whether the domain was registered more than 12 months ago. If it was, the heuristic deems it a legitimate site. Otherwise it deems it a phishing site. A shortcoming of this heuristic was that newly created legitimate sites are not registered in one year. In this case, the heuristic will fail. Another shortcoming is that domain names of many phishing sites were in fact registered over a year ago. Especially, modern phishing sites are often discovered on a host owned by legitimate company. Some vulnerability in that host allowed a phisher to penetrate it and set up a phishing sites. In such cases, the domain name was often registered long time ago, and thus, the heuristic fails to classify it correctly.

4.2.2. Known Images (H_2)

This is a check of whether a page contains inconsistent use of well-known logos such as those of eBay, PayPal, Citibank, Bank of America, Fifth Third Bank, Barclays Bank, ANZ Bank, Chase Bank, and Wells Fargo Bank.

For instance, if a site contains eBay logos but is not on an eBay domain, the heuristic deems it a phishing site. However, the function of pattern-matching in a digitized image might lead to many misjudgments. In the other case, this heuristic also fails when legitimate sites employ these logo files. Even if a company has a business relationship with PayPal and uses the PayPal logo in its website, the heuristics labels this as a phishing site.

4.2.3. Suspicious URL (H_3)

This is a check of whether the site URL contains an at-mark (@) symbol or a hyphen (-) in the domain name. If so, the heuristic deems it a phishing site because phishing attackers are likely to use these symbols in their domain name of a phishing site. The weakness of the heuristics are that some legitimate sites, (e.g., aist-nara.ac.jp), use a hyphen in their domain name. Several phishing sites also do not have an at-mark or a hyphen in their domain name.

4.2.4. Suspicious Links (H_4)

Similar to the Suspicious URL heuristic, this one checks if a link on the page contains an at-mark or a hyphen. The weak points of this heuristic are same as those of the Suspicious URL heuristic.

4.2.5. IP Address (H_5)

This is a check of whether the domain name of the site is an IP address. Though legitimate sites rarely link to pages via an IP address, phishers often attract victims to phishing sites by IP address links. The heuristic fails if the URL of a phishing site uses a fully qualified domain name, or that of a legitimate site is an IP address.

4.2.6. Dots in URL (H_6)

This is a check of whether the URL of the site contains five or more dots. According to Fette *et al.* [6], dots can be abused for attackers to construct legitimate-looking URLs. One technique is to have a sub-domain. Another is to use a redirection script, which to the user may, for instance, appear like a site hosted at google.com, but in reality will redirect the browser to phishing.com. In both of these examples, either by the inclusion of a URL into an open redirect script or by the use of a number of sub-domains, there are a large number of dots in the URL. The heuristic fails if there are fewer than five dots in the URL of a phishing site. For instance, a phishing site, which was reported November 2008 and placed at *http://kitevolution.com/os/chat6/plugins/safehtml/www.paypal.com/canada/cgi-bin/webscr.php?cmd=_login-run*, includes only four dots. Conversely, the URLs of some legitimate sites can have five or more dots.

4.2.7. Forms (H_7)

This is a check of whether the page contains web input

forms. It scans the HTML for <input> tags that accept text and are accompanied by labels such as "credit card" and "password". If this is the case, the heuristic deems it a phishing site. Unfortunately, this heuristic fails in labeling whenever phishers uses digital images of such words rather than using actual text.

4.2.8. TF-IDF-Final (H_8)

This is a check of whether the site is phishing by employing TF-IDF-Final, an extension of the Robust Hyperlinks algorithm [14]. When the heuristic attempts to identify phishing sites, it feeds the mixture of word lexical signatures and the domain name of the current web site into Google. If the domain name matches the domain name of the top 30 search results, the web site is labeled legitimate. Some phishing sites, however, can be made to rank more highly in search results by manipulation of the search result page.

4.3. Experimental Design

We used a within-subjects design, where every participant saw every website and judged whether or not it appeared to be a phishing site. In our test we asked 10 participants to freely browse the websites. Each participant's PC was in-stalled with Windows XP and Internet Explorer (IE) version 6.0 as the browser. Other than configuring IE to display IDN, we installed no security software and/or anti-phishing toolbars. We also did not prohibit participants from accessing websites not listed in **Table 1**. Some participants therefore inputted several terms into Google and compared the URL of the site with the URLs of those listed in Google's search results.

In this experiment, we used the average error rate as a performance metric. To average the outcome of each test, we performed 4-fold cross validation and repeated in 10 times. However, we considered that the experiment involved a small, homogeneous test population; therefore it would be difficult to generalize the results toward typical phishing victims. We will discuss our plan for removing the bias in Section 6.

4.4. Experiment Results

First, we invited 10 participants, all Japanese males, from the Nara Institute of Science and Technology. Three had completed their master's degree in engineering within the last five years, and the others were master's degree students. We let participants to label the websites described in **Table 1**. The results by each participant are shown in **Table 2**. A hash mark (#) denotes the number of websites in **Table 1**, P_1 - P_{10} denote the 10 participants, "F" denotes that a participant failed to judge the website, and the empty block denotes that a participant succeeded in judging it correctly.

Next, we determined the detection accuracy of the AdaBoost-based detection method. We used eight heuristics and outputted a binary variable representing phishing or not-phishing. The detection results by each heuristic are shown in **Table 2**, where H_1 - H_8 denote eight heuristics in which numbers are correspond to Section 4.2.

Finally, we measured the detection accuracy of HumanBoost. We constructed 10 PTD databases. In other

Table 2. Detection results by each participant and heuristic, in November 2007.

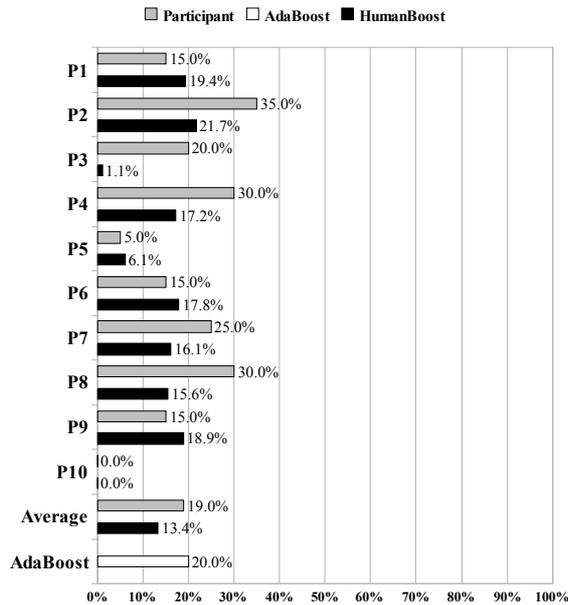| #	Participants										Heuristics							
	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8
1	-	-	-	-	-	-	F	F	-	-	-	-	-	-	-	F	-	-
2	F	F	-	-	-	-	-	-	-	-	-	-	-	F	F	F	F	-
3	-	F	-	-	-	-	-	-	-	-	F	F	-	F	F	F	-	-
4	-	-	F	-	-	F	-	-	F	-	-	F	-	-	-	-	-	-
5	-	-	-	-	-	-	F	F	-	-	-	-	F	-	F	-	F	-
6	-	F	F	-	F	F	-	-	-	-	-	F	F	F	F	F	F	-
7	F	F	-	F	-	F	-	-	-	-	-	F	-	-	-	-	F	F
8	-	-	-	-	-	-	-	-	-	-	-	-	-	F	F	F	F	-
9	-	-	-	-	-	-	-	-	-	-	F	F	-	F	F	F	-	-
10	-	-	-	F	-	-	-	-	-	-	-	F	F	-	-	F	-	-
11	-	-	-	-	-	-	F	-	-	-	F	F	F	F	F	F	-	-
12	-	-	-	-	-	-	F	-	F	-	-	F	-	-	-	-	F	F
13	-	F	F	-	-	-	-	-	-	-	-	-	-	-	-	-	F	-
14	-	-	-	-	-	-	-	F	-	-	F	-	F	F	F	F	F	-
15	-	-	-	F	-	-	-	-	-	-	-	F	-	-	F	F	-	-
16	-	F	-	F	-	-	-	F	-	-	F	F	F	F	F	F	-	-
17	-	F	-	F	-	-	-	F	-	-	-	F	F	F	F	-	F	-
18	-	-	-	-	-	-	-	-	-	-	-	-	F	F	-	-	F	-
19	F	-	-	F	-	-	F	F	F	-	-	F	-	-	-	-	F	-
20	-	-	F	-	-	-	-	-	-	-	-	-	-	F	F	F	F	-

Figure 2. Average error rates of each participant, adaboost-based detection method, and HumanBoost in the pilot study, in November 2007.

words, we made 10 types of 20 * 9 binary vectors. Under the same conditions described above, we calculated the average error rate for each case of HumanBoost.

The results are summarized in **Figure 2**, where the gray bars denote the error rate of each participant, the white bar denotes the average error rate of the AdaBoost-based detection method, and the black bars denote that of HumanBoost. The average error rate for Human-Boost was 13.4%, 19.0% for the participants and 20.0% for the AdaBoost-based detection method. The lowest false positive rate was 19.6% for HumanBoost, followed by 28.1% for AdaBoost and 29.7% for the participants. The lowest false negative rate was 8.5% for HumanBoost, followed by 13.5% for AdaBoost, 14.0% for the participants.

We found that the average error rate of some participants increased by employing HumanBoost. We analyzed the assigned weights and found that some heuristics were assigned higher weights than such users' trust decision. For instance, participant 9 had labeled three legitimate sites as phishing sites, whereas the existing heuristics had labeled these three sites correctly. His trust detection was therefore inferior to that of existing heuristics and we assumed that this is the reason for the increase in error rate.

5. Follow-up Study

Increasing the number of participants essentially enables us to generalize the outcome of HumanBoost. In this section, we explain the two cases of the follow-up stud-

ies performed in 2010. Note that the pilot study was performed in November 2007 and the follow-up studies were performed in March 2010 and July 2010, therefore may be difference based on the demographics of the participants and substantial media coverage about phishing.

5.1. A Case of the Follow-up Study in March 2010

Our follow-up study had 11 new participants, aged 23 to 30. All were from the Japan Advanced Institute of Science and Technology. All were Japanese males, two had completed their master's degree in engineering within the last five years, and the others were master's degree students.

Before conducting the follow-up study, we modified the dataset described in **Table 1**. Due to the renewal of PayPal's website during 2007 - 2010, we updated websites 9 and 20 to mimic the current PayPal login pages. Particularly, Nanto Bank, website 6 in **Table 1**, had changed both the URL and the content of its login page. Nanto Bank is also not well-known in Ishikawa Prefecture, where the participants of the follow-up study lived. We therefore changed website 6 to Hokuriku Bank (another Japanese regional bank in Ishikawa). The domain name of Hokuriku Bank is *www2.paweb.answer.or.jp*, the same as Nanto Bank.
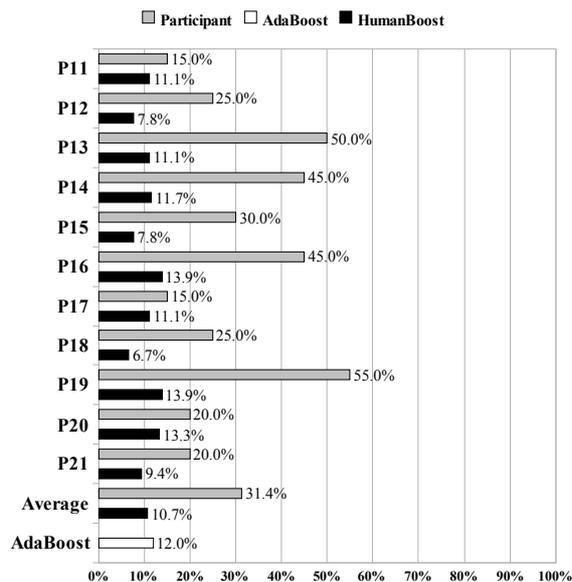
In March 2010, invited 11 participants and asked them to label 20 websites as legitimate or phishing. Different from the pilot study described in Section 4, we prepared printed documents to expedite this experiment. Instead of operating a browser, participants looked at 20 screen shots of a browser that had just finished rendering each website. Additionally, showing a browser screen shot is often used for phishing IQ tests.

The detection results by each participant and each heuristic are shown in **Table 3**. A hash mark denotes the number of websites, P_{11} - P_{21} denote the 11 participants, H_1 - H_8 denote the eight heuristics, "F" denotes that a participant or heuristic failed to judge the website, and the empty block denotes that a participant or heuristic succeeded in judging correctly. We also calculated the average error rate for each participant, the AdaBoost-based detection method, and HumanBoost.

The results are shown in **Figure 3**, where the gray bars denote the error rate of each participant, the white bar denotes the average error rate of the AdaBoost-based detection method, and the black bars denote that of HumanBoost. The lowest error rate was 10.7% for Human-Boost, followed by 12.0% for AdaBoost and 31.4% for the participants. The lowest false positive rate was 15.4% for AdaBoost, followed by 18.1% for HumanBoost and 39.9% for the participants. The lowest false negative rate was 6.1% for HumanBoost, followed by 8.4% for AdaBoost and 25.9% for the participants. In comparison

Table 3. Detection results by each participant and each heuristic in the follow-up study, in March 2010

#	Participants											Heuristics							
	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	P_{17}	P_{18}	P_{19}	P_{20}	P_{21}	H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8
1	-	-	-	-	-	F	-	-	-	F	F	-	-	-	-	-	F	F	-
2	-	F	-	-	-	-	-	F	-	-	-	-	F	-	F	F	F	F	-
3	-	-	F	F	F	-	-	-	F	-	-	F	F	-	F	F	F	-	-
4	F	-	-	-	-	-	-	F	F	-	-	-	-	-	-	-	F	F	-
5	-	-	F	-	-	-	-	-	-	-	-	-	F	F	-	F	-	F	-
6	-	-	-	F	-	-	F	-	F	-	-	-	F	F	F	F	F	F	-
7	-	F	F	F	F	-	F	F	-	-	F	-	-	-	-	-	-	F	F
8	-	-	F	-	-	F	-	-	F	-	-	-	F	-	F	F	F	F	-
9	-	-	-	-	-	-	-	-	-	-	-	F	F	-	F	F	F	-	-
10	F	-	-	-	-	F	-	-	F	-	-	-	F	F	-	-	F	-	-
11	-	-	F	F	-	-	-	-	F	-	-	F	F	F	F	F	F	-	-
12	-	-	-	-	-	-	-	F	F	-	-	-	F	-	F	-	-	F	-
13	-	F	F	-	F	-	-	F	F	-	F	-	F	-	-	-	-	F	-
14	-	-	-	-	-	F	-	-	F	F	-	F	F	F	F	F	F	F	-
15	-	F	-	F	F	-	-	-	-	-	-	-	F	-	-	F	F	-	-
16	-	F	F	F	F	F	F	F	-	F	-	-	F	F	F	F	F	-	-
17	-	-	F	F	-	F	-	-	F	-	-	-	-	F	F	F	-	F	-
18	-	-	-	-	-	F	-	-	-	-	-	-	F	F	F	-	-	F	-
19	F	-	F	F	F	F	-	-	-	-	F	-	F	-	-	-	-	F	-
20	-	-	F	F	-	F	-	-	F	-	-	-	F	-	F	F	F	F	-

Figure 3. Average error rates of each participant, adaboost-based detection method, and HumanBoost in the follow-up study, in March 2010

to the pilot study, the average error rate in participants increased due to the difference in the experimental design; the pilot study allowed participants to operate a browser but the follow-up study did not. However, we observed that HumanBoost achieved higher detection accuracy.

5.2. A Case of the Follow-up Study in July 2010

In order to collect more users' PTDs, we recruited participants via Internet research company. In this section,

we summarize the results briefly.

Of the recruited 309 participants, 42.4% (131) were male and 57.6% (178) were female. Age ranged from 16 to 77 years old. 48.2% of participants (149) were office workers, and 19.7% (61) were households and 5.8% (18) were students. Of the students, 66.7% (12) were Bachelors, 11.1% (2) were high school students, 5.6% (1) was a master's degree student. They mainly lived around Tokyo area. We therefore changed website 6 to Tokyo Tomin Bank (another Japanese regional bank in Tokyo). The domain name of Tokyo Tomin Bank is also *www2.paweb.answer.or.jp*. The other conditions of this study are the same as the follow up study described in Section 5.1. In July 2010, recruited 309 participants looked at 20 screen shots and judged whether the site seems to be phishing or legitimate.

Based on the detection results, we also calculated the average error rate for each participant, the AdaBoost-based detection method, and HumanBoost. The lowest error rate was 9.7% for HumanBoost, followed by 10.5% for AdaBoost and 40.5% for the participants. The lowest false positive rate was 18.3% for AdaBoost, followed by 19.5% for HumanBoost and 57.4% for the participants. The lowest false negative rate was 5.5% for HumanBoost, followed by 7.1% for AdaBoost and 33.2% for the participants.

6. Discussion

6.1. Comparative Study with SVM

As mentioned in Section 2, the limited numbers of heuristics is one of the biggest issues on detecting phishing sites. We attempted to utilize users' PTDs as a new heu-

ristic and incorporated it into existing heuristics by using AdaBoost. Though AdaBoost has two advantages as explained in Section 3.2, checking whether the PTDs are useful for other machine learning techniques is necessary.

In this section, we employ Support Vector Machine (SVM), which is also one of the typical machine learning techniques for classification. We used SVM to incorporate heuristics instead of AdaBoost, and calculated the average error rate in the cases of with and without users' PTDs. To clarify our explanation, we named the method of using eight existing heuristics with SVM as SVM-based detection method. We also named HumanSVM, the method of incorporating PTDs into existing heuristics.

First, we calculated the average error rates by using the dataset in the pilot study, as described in Section 4. The conditions are the same as the pilot study, excepting a machine learning method. The results showed that the average error rate for HumanSVM was 14.3% and that for SVM-based detection method was 21.7%.

Second, we used the dataset in the follow-up study performed in March 2010, described in Section 5.1. The results showed that for HumanSVM was 11.4% and that for SVM-based detection method was 18.3%.

Finally, we calculated the average error rates by using the dataset in the follow-up study performed in July 2010, described in Section 5.2. The result showed that the average error rate for HumanSVM was 11.2%, for SVM-based detection method was 18.9%.

We observed that the average error rates were decreased by using PTDs in the all cases. We also observed that the average error rates in HumanSVM (14.3%, 11.4%, and 11.2%) were higher than that in HumanBoost (13.4%, 10.7%, and 9.7%). Albeit HumanBoost performed better than HumanSVM, we assumed that the utilization of PTDs is available as a new heuristic for detecting phishing sites.

6.2. Removing the Bias

In this section, we discuss our plan for removing the bias. Removing bias is generally important for a participant-based test. Though we used cross validation, the presence of bias can still be assumed due to the biased dataset and/or biased samples.

Especially, we assumed that labeling our prepared websites was much difficult than labeling the typical phishing websites and/or legitimate sites. As explained in Section 4.1, we designed our study referred to the typical phishing IQ tests. Since the almost of our prepared websites contained traps, participants often failed to label the sites. These traps also hindered the existing heuristics to classify websites. It might result the average error rates

remained still higher.

We positioned our laboratory tests as a first step, and decided to perform a field test in a large-scale manner. One approach toward field testing is implementing a HumanBoost-capable phishing prevention system. This is possible by distributing it as browser-extension with some form of data collection and getting a large population of users to agree to use it.

6.3. Issues on the Implementation of HumanBoost-Capable Systems

Here we consider some issues that arise in implementing a HumanBoost-capable system. Imagine if HumanBoost has been available in phishing-prevention systems.

The HumanBoost-capable system's weak point is that always works after the user finishes making a trust decision. Generally, phishing-prevention systems are to prevent users from visiting phishing sites. Apart from these systems, HumanBoost requires users to judge if their confidential information can be input to the site.

Another problem is difficulty in convincing users to reconsider their trust decisions. When users attempt to browse a phishing site, typical phishing prevention systems display an alert message. In HumanBoost, such messages would be shown after making the trust decision. If the user relies on his/her trust decision, the HumanBoost-capable system will not work if the system alerts correctly.

To solve these problems, the HumanBoost-capable system should have the ability to cancel the input or the submission of users' confidential information, instead of blocking the phishing site. The system should monitor such events, e.g., inputting any data to input forms and/or clicking buttons. The system should also hook these event handlers not to send any information if the site deems to be phishing. It is possible to implement such system as a browser-extension, as mentioned in Section 6.2.

The HumanBoost-capable system should also have an interface that can expedite users re-making trust decisions. For instance, the system shows an alert window by interrupting users' browsing. The alert window should contain some text which convince user that the reason of the site seems to be phishing. It is also possible to implement such system as a browser-extension.

7. Conclusions

In this paper, we presented an approach called HumanBoost to improve the accuracy of detecting phishing sites. The key concept was utilizing users' past trust decisions (PTDs). Since Web users may be required to make trust decisions whenever they input their personal information into websites, we considered recording these trust deci-

sions for learning purposes. We simply assumed that the record can be described by a binary variable, representing phishing or not-phishing, and found that the record was similar to the output of the existing heuristics.

As our pilot study, in November 2007, we invited 10 participants and performed a subject experiment. The participants browsed 14 simulated phishing sites and six legitimate sites, and judge whether or not the site appeared to be a phishing site. We utilized participants' trust decisions as a new heuristic and we let AdaBoost incorporate it into eight existing heuristics.

The results showed that the average error rate for HumanBoost was 13.4%, whereas that of participants was 19.0% and that for AdaBoost was 20.0%. We also conducted the follow-up study in March 2010. This study invited 11 participants, and was performed in the same fashion of the pilot study. The results showed that the average error for HumanBoost was 10.7%, whereas that of participants was 31.4%, and that for AdaBoost was 12.0%. Finally, we invited 309 participants and performed the follow-up study in July 2010. The results showed that the average error rate for HumanBoost was 9.7%, whereas that of participants was 40.5% and for AdaBoost was 10.5%. We therefore concluded that PTDs are available as new heuristics and HumanBoost has the potential to improve detection accuracy for Web user.

We then checked if PTDs are useful for another machine learning-based detection method. For a case study, we employed SVM and measured detection accuracy in the cases of with and without PTDs. The results showed that the utilizing PTDs increased the detection accuracy.

We therefore concluded that the PTDs are available as new heuristics and HumanBoost has the potential to improve detection accuracy for Web user.

8. Acknowledgements

REFERENCES

[1] T. McCall and R. Moss, "Gartner Survey Shows Frequent Data Security Lapses and Increased Cyber Attacks Damage Consumer Trust in Online Commerce," 2005. http://www.gartner.com/press_releases/asset_129754 _11.html

[2] C. Pettey and H. Stevens, "Gartner Says Number of Phishing Attacks on U.S. Consumers Increased 40 Percent in 2008," April 2009. http://www.gartner.com/it/page.jsp?id=936913

[3] Anti-Phishing Working Group, "Phishing Activity Trends Report-Q1, 2008," August 2008. http://www.apwg.com/reports/apwgreport_Q1_2008.pdf, 0

[4] Y. Zhang, S. Egelman, L. Cranor and J. Hong, "Phinding Phish: Evaluating Anti-Phishing Tools," *Proceedings of the* 14*th Annual Network and Distributed System Security Symposium*, USA, February 2007.

[5] D. Miyamoto, H. Hazeyama and Y. Kadobayashi, "An Evaluation of Machine Learning-Based Methods for Detection of Phishing Sites," *Australian Journal of Intelligent Information Processing Systems*, Vol. 10, No. 2, 2008, pp. 54-63.

[6] I. Fette, N. Sadeh and A. Tomasic, "Learning to Detect Phishing Emails," *Proceedings of the* 16*th International Conference on World Wide Web*, Canada, May 2007, pp. 649-656.

[7] S. Abu-Nimeh, D. Nappa, X. Wang and S. Nair, "A Comparison of Machine Learning Techniques for Phishing Detection," *Proceedings of the* 2*nd Annual Anti-Phishing Working Groups eCrime Researchers Summit*, USA, October 2007, pp. 60-69.

[8] R. Basnet, S. Mukkamala and A. H. Sung, "Detection of Phishing Attacks: A Machine Learning Approach," *Studies in Fuzziness and Soft Computing*, Vol. 226, February 2008, pp. 373-383.

[9] Y. Pan and X. Ding, "Anomaly Based Web Phishing Page Detection," *Proceedings of the* 22*nd Annual Computer Security Applications Conference on Annual Computer Security Applications Conference*, USA, September 2006, pp. 381-392.

[10] Y. Zhang, J. Hong and L. Cranor, "CANTINA: A Content-Based Approach to Detect Phishing Web Sites," *Proceedings of the* 16*th World Wide Web Conference*, China, May 2007, pp. 639-648.

[11] Open DNS, "Phishtank-Join the Fight against Phishing," http://www.phishtank.com.

[12] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Science*, Vol. 55, No. 1, August 1997, pp. 119-137.

[13] R. Dhamija, J. D. Tygar and M.A. Hearst, "Why Phishing Works," *Proceedings of Conference on Human Factors in Computing Systems*, April 2006, pp. 581-590.

[14] A. Y. Fu, X. Deng, L. Wenyin and G. Little, "The Methodology and an Application to Fight against Unicode Attacks," *Proceedings of the* 2*nd Symposium on Usable Privacy and Security*, USA, July 2006, pp. 91-101.

[15] T. A. Phelps and R. Wilensky, "Robust Hyperlinks: Cheap, Everywhere, Now," *Proceedings of the* 8*th International Conference on Digital Documents and Electronic Publishing*, September 2000, pp. 28-43.