

# CUSP: Customizable and Usable Spam Filters for Detecting Phishing Emails

Madhusudhanan Chandrasekaran, Vidyaraman Sankaranarayanan, Shambhu Upadhyaya  
Department of Computer Science and Engineering,  
University at Buffalo, State University of New York, Buffalo, NY 14260  
{mc79, vs28, shambhu}@cse.buffalo.edu

**Abstract**—Phishing attack continues to be a significant threat to the Internet users and commercial organizations worldwide causing billions of dollars in damage. A successful phishing attack depends on the inability of an end user to accurately tell legitimate and spoofed emails apart. However, unlike their legitimate counterpart, as spoofed emails are composed in bulk, they do not contain any user specific data, which relates users with their accounts. In this paper, as a first step, we propose a customizable spam filter that allows the users to store this user specific data on a per organization basis, and then use the stored data to discriminate against fraudulent emails. As a next step, we propose a NLP based technique to generate context sensitive warnings that would help in educating users about the dangers of phishing attack. Lastly, we test and validate our framework on existing phishing corpus and live emails.

**Index Terms**—Context-sensitive warnings, Email Fraud, FrameNet, Natural Language Processing (NLP), Phishing, WordNet

## I. INTRODUCTION

Phishing is a Web based attack where attackers trick users into revealing confidential information such as password, credit card number, social security number (SSN), or bank account numbers in fraudulent Websites that mimic the look-and-feel of their legitimate counterpart. This redirection of users into fraudulent Websites can be achieved either through social engineering attacks or forcefully using malware. Even though a variety of threat vectors such as instant message spamming (spimming), public forum spamming and DNS redirection (pharming), can be used for phishing, due to its widespread adoption and ability to be easily spoofed, email continues to be the preferred vehicle to launch such attacks. Recent studies estimate that in 2007 alone, more than 25,000 unique phishing emails hijacking 150 different brands were sent out on a per month basis, resulting over \$3 billion dollars in damage worldwide [9].

Given the significant impact on global economy, several anti-phishing efforts have been undertaken in both academia and industry to detect and mitigate phishing attacks. Most of these approaches are implemented as browser add-ons and third-party toolbars that operate on every URL visited by a user to determine their authenticity. Despite their initial success in protecting the users from divulging confidential information across fake Websites,

they have several drawbacks. First of all, most of these toolbars adopt decisions based on blacklists (or whitelists), which are propagated to them by some centralized servers. However, as phishing sites are ephemeral, the probability that these lists reach the clients in time is very low. Second, a recent study on 10 popular anti-phishing toolbars conducted by Zhang et al. [15] showed that the toolbars when tested on live phishing data, exhibit poor performance having an overall accuracy of less than 60%. Lastly, as these toolbars operate close to the source of the attack (i.e., on Websites rather than emails), any misclassification error on their part would imply that their users are left defenseless. An alternative to detect phishing attack is to filter out spoofed emails before they reach user's mailbox. Traditionally, anti-spam mechanisms were used for this purpose. Although phishing emails can be regarded as *unsolicited junk*, they do not share the same characteristics as spam emails, thereby requiring specialized filters for classification. In this context, a few specialized efforts have been undertaken that attempt to classify phishing emails based on the features intrinsic to them: such features include, but are limited to, the content type of the message (Plain text/HTML), nature of the contained URLs (dotted IP/encoded format), credibility of the referred domains, words that frequently appear in the phishing email content, etc. However, due to the instant availability of automated tools, it has become possible for the phishers to fabricate phishing emails just by using a reduced subset of these features that can evade even the sophisticated email filters. Moreover, as the features used by these approaches (e.g., frequently occurring words, different visible and referred-to URLs, number of URLs, and email MIME type) are also present in the emails sent by the legitimate institutions, it has become extremely hard to build generic classifiers that can accurately detect phishing emails. To overcome these limitations, we adopt a proactive approach, which attempts to detect phishing emails based on the user specific data contained in them. The main assumption here is that as phishing emails are composed in bulk, they lack in any data that can relate the users to their personal information; on the contrary, legitimate financial institutions send out directed emails to customers using personalized data that are not known publicly (transaction identifiers (tids), abbreviated version of their account number, full name, date-of-birth, address, etc.) This private data, in turn, can act as shared

authentication secret used to validate the sending domain's legitimacy.

In this paper, as our first contribution, we propose CUSP, a Customizable and Usable Sпам filter to detect Phishing attacks, which allows users to store private data on a per organization/account basis. Subsequently, every incoming email that purports to originate from the stored organization is examined to see if it contains the previously stored data. If there is a mismatch (or the data is absent), the email is deemed as suspicious. The notion of verifying the sender's domain for detecting spoofed emails is not new; there exists mechanisms like SPF (Sender Policy Framework), Sender ID, DKIM (Domain Key Identified Mail) that validate the sending domain using IP addresses or digital signatures. Although these mechanisms can vouch for the sending domain's reputation, they still fail to stop users from falling prey to phishing attacks. Furthermore, unlike CUSP, these mechanisms are heavyweight – each of them adopts a different protocol that requires changes to the existing email infrastructure. For the next step, we focus on generating context-sensitive warnings that help users in identifying phishing attacks. As phishing is a social engineering attack, their emails falsely impose an implied sense of urgency and threat (account suspension) or lure and cajole (reward for completing a survey) to trick the users into visiting fake Websites. As our second contribution, we propose a novel technique that relies on context-sensitive text categorization as a means to detect the “tone” or the implied message of the email. We consider this identification a critical factor towards not only identifying the phishing emails, but also communicating the import of the email to the end user. Consider the phishing emails that get past the standard phishing filter: if our framework can provide a meaningful communication to the user regarding the intentions of the email sender, it would not only be an effective methodology to defeat the attack, but could also educate the naïve user against the potential harmful effects, which, after all, is the key to defeating these attacks. We implement CUSP as a plug-in to Microsoft Outlook – a popular email client used by both home and corporate users. Lastly, we evaluate CUSP against existing corpus and report our findings.

The rest of this paper is organized as follows. We begin Section 2 by discussing related work. Section 3 presents a short survey of user specific data contained in financial emails from institutions that are vulnerable to phishing attacks. Section 4 presents the threat model that we are targeting to address in this paper. Section 5 presents an overview of CUSP and its implementation details. The generation of context-sensitive warnings is also discussed here. Section 6 presents the results of our evaluation and discusses the limitations of CUSP. In Section 7, we conclude the paper.

## II. RELATED WORK

In this section, in order to bring out the efficacy of our approach, we briefly compare and contrast our work with other related approaches.

### A. Browser Plug-ins and Anti-Phishing Toolbars

Since most of the phishing attacks rely on the inability of users to discern legitimate and fake emails apart [17], several commercial and open source toolbars have been proposed to assist the users in determining the validity of the visited Websites. Spoofstick [14] is one such browser add-on which displays the IP address of the spoofed URLs in its toolbar. As it requires the end user to discriminate between the fake and the real Websites, it does not provide an automated solution to detect phishing attacks. NetCraft antiphishing toolbar [12] is another monitoring tool that employs client-server architecture. Each toolbar subscriber acts as a client and is responsible for reporting suspicious Websites to a central server. The server then processes every incoming request by checking the domain age, hosted location and URLs, and then puts the reported Websites either into a whitelist or a blacklist. These lists are propagated to other clients to assist them with their decision making. A disadvantage with such an approach is that as phishing Websites are ephemeral, it might not be possible to propagate the generated blacklists to the clients in time. SpoofGaurd [6] is a browser plug-in that examines the visited Website using stateful and stateless evaluation. The stateless evaluation includes check for invalid links, URL obfuscation attacks, valid https connection and authenticity of SSL/TLS certificates. It also checks to determine whether the images present in the legitimate sites are imported by unknown suspicious domains. Unlike stateless evaluation, stateful page evaluation monitors every outgoing data using site specific salts so that a user does not provide his username and password into a site he has never visited before. In most cases, the final result of these toolbars is either binary (phishing or safe) or ternary process (where a score/color is displayed on the toolbar to warn users about the sites' suspiciousness.) Despite their advantages, a recent study [15] experimented with 10 popular anti-phishing toolbars revealed that the toolbars failed to identify 15% of the phishing Websites used for testing. Also, as discussed earlier, these toolbars depend on the validity of IP as an important detection criterion and fail to protect from attacks launched from the legitimate Websites. Lastly, these toolbars ignore the weak human factor and require users to make the final decision, i.e., to trust a suspicious Website or not.

### B. Digital Signing and PKI Based Schemes

Digital signing and trust propagation schemes have been proposed to make email secure and reliable. These schemes employ publicly available standards such as S/MIME, PGP and GPG to encrypt, decrypt and validate email messages. Spam protection framework (SPF), Certified Sender Validation (CSV) and DomainKeys have

also been proposed as an alternative mechanism to authenticate emails based on their sender's domain name. DomainKeys uses digital signatures to authenticate domain name and the entire content of a message, whereas SPF and CSV look at the email headers to identify forgery. Even though these schemes act as an effective anti-spoofing solution, they suffer from several disadvantages. First, adoption of these techniques necessitates steep learning curve which might be elusive to everyday users. Second, these techniques require installation of additional software to support S/MIME, PGP, GPG, etc. These provisions are not readily available in most of the popular Web based email clients such as Yahoo Mail, Hotmail and Gmail. Finally, these techniques suffer from key distribution problems, where a trusted medium is needed to exchange keys needed to sign and encrypt/decrypt messages. In the case of PGP/GPG schemes, as there is no central authority server, a phisher can infiltrate the Web of trust by digitally signing his emails. Another drawback of this PKI-based and authentication based approaches is that both the sender and the receiver need to have the same signing and verification mechanisms.

### C. Content based Phishing Attack Detection

Several research efforts employing machine learning and pattern recognition techniques have been proposed to classify phishing emails. Most of the earlier algorithms were tailored to detect spam emails and did not perform well when applied in the context of phishing attacks. These approaches were naïve in the sense that they essentially focus on detecting the presence of uncommon words that appear in the spam emails [5]. As phishing emails closely imitate their legitimate counterpart, unlike spam, they do not contain such random and junk words. In order to classify phishing emails, Fette et al. [8] employ a set of 16 different machine learning algorithms operating on a predefined feature set. The feature set consists of structural elements that indicate presence of illegitimate hyperlinks, IP based URLs, non-matching URLs and other characteristics intrinsic to phishing emails. CANTINA [16] is another tool which uses term frequency and inverse document frequency (tf-idf) to identify commonly appearing words in phishing Web pages. These words along with other structural elements are used as features for classification. As opposed to these heavy-weight approaches, CUSP assists the users to filter out phishing emails based on the user-specific data contained in them. Also, based on the tone of the phishing email, context-sensitive warnings are generated to let the user know the working of phishing attacks.

### III. USER SPECIFIC DATA IN THE EMAILS FROM LEGITIMATE INSTITUTIONS: A BRIEF SURVEY

In order to demonstrate the feasibility of our approach, we present a brief survey on the user specific data contained in the emails from legitimate institutions. This would also allow us to identify the private data that need to be stored in CUSP so that accurate prediction of phishing emails is

possible. For this survey, we consider the top 20 most phished brands in 2007, as reported by Phishtank. Phishtank, a collaborative undertaking of academia and industry, operates by assimilating and publicizing phishing email feeds, which are then verified by the interested subscribers. Out of these 20 brands, 17 are online banks and credit card institutions. The remaining three are popular Internet portals that support e-business. The summary of our findings are presented in the form of a table (see Appendix A). All the 20 brands claim that they do not send emails to the customers requesting their personal credentials. Furthermore, the banks' Websites clearly state that any email carrying such information on their behalf is a fraudulent one. Majority of the banks also claim to send out personalized emails to the customers (i.e., having information such as their last/full name, last four digits of their account number, and occasionally their home address.) However, there were mixed response on whether such data can be used for validation purposes. While most of the banks advised the customers to use this data as one of the "visual indicators" to identify spoofed emails, one bank cautioned otherwise citing "spear phishing" as the example. It is important to note that even though it may be possible for an attacker to launch targeted phishing attacks (spear phishing) by using the recipients' private data obtained through other means, they are usually rare due to the difficulty involved. A recent study involving real human subjects shows that the users place implied trust on personalized emails [10]. Although the underlying intention was correct, the subjects were not able to make a clear distinction on whether the personalized data is actually the private data (i.e., not publicly known). For example, the subjects incorrectly trusted the emails that contained first four digits of the credit card number, even though first four digits are not random and are dependent on the card issuer.

### IV. THREAT MODEL

In this paper, we restrict our focus to sifting legitimate emails and phishing emails based on user specific data contained in them. Even though it might be possible for an attacker to acquire the user specific data through "dumpster-diving" or illegitimately accessing the user's mailbox, nevertheless such targeted approaches are not scalable from the attacker's standpoint. Phishing attacks can be enforced via different mediums such as "chat", "phone", or by using malware. The defenses against these threats, however, are beyond the scope of our paper. As is, the main shortcoming of our approach is that it cannot be used in cases where the institution under consideration does not include personalized content in their emails. Popular email services like Yahoo, Gmail cannot address users with any personal information. Even in companies like Amazon.com and PayPal, as there is no concept of user account number, the only identifying data available is the user's name. Obtaining the user's name is relatively easy, when compared to other private data such as last four digits of the account number. Hence, in such cases, it may be possible for an attacker to

evade CUSP by using spear phishing attacks with emails comprising of this publicly available user information.

## V. OVERVIEW OF CUSP

In this section, we describe how CUSP can be used to detect fraudulent emails from known institutions by giving out its working details. CUSP has been developed as a plug-in for Microsoft Outlook in C# using Visual Studio Tools for Office 2003 (VSTO). CUSP attaches itself to the email client, and is bootstrapped with a list of popular institutions that are prone to phishing attacks. At the time of installation, if a user is subscribed with any of the preloaded institutions contained in CUSP, then he is required to specify the corresponding user specific data that are to be included in legitimate emails from them. Figure 1 shows a form in CUSP requesting such information from the user. In case if an institution of user's choice is not present in CUSP, then he can add a custom tag to include it. Similarly, as there is no common consensus among institutions on what user specific data are to be included in their emails, the user is also provided with an option to add/modify the existing tags representing different fields such as his address, product key, date-of-birth, etc. We also understand that it might be difficult for a naïve user to figure out beforehand the data to be included in CUSP corresponding to a given institution. Ideally, such information needs to be updated by the software provider, as opposed to the user. The user specific data are hashed and stored in CUSP similar to the way in which values for auto-completion fields are stored in a browser. This also ensures that any compromise of CUSP does not easily give away the user information.

Fig. 1. CUSP requesting the user to enter private data corresponding to the subscribed institution

Any email that purports to originate from the preloaded organizations is examined to see if it contains the relevant user specific data. If it does, then the email is tagged “safe”

and sent to the user’s mailbox. On the other hand, if the user specific data is missing or is incorrect, the email is tagged as “phishing.” If a user fails to enter the required information at the time of installation, a dialog box is prompted asking for relevant information as shown in Figure 2. The user, also, has an option of regarding the email message as not a financial institution. Using this option indiscriminately exposes risk of the user falling prey to phishing attack. If an email is tagged as phishing, then the content of the email is analyzed to extract the “tone” conveyed in the email so that appropriate context-sensitive warning messages can be generated. The process of generating appropriate context-sensitive warnings is discussed in the next section. Once the warnings are generated, they are communicated back to the user in a text box. Figure 3 shows the text box indicating warning messages for a HSBC phishing email which threatens users to disclose sensitive information by using account revocation as an argument. For a user who is still not convinced by the warning, an option of forwarding the email to the provided security department is provided so that accurate response about the validity of the email can be obtained. Disallowing the user to respond to the email until proper clearance is obtained can also be done in a forceful manner as shown in [3].



Fig. 2. A modal box interrupting the user to enter the required user specific data before opening the email

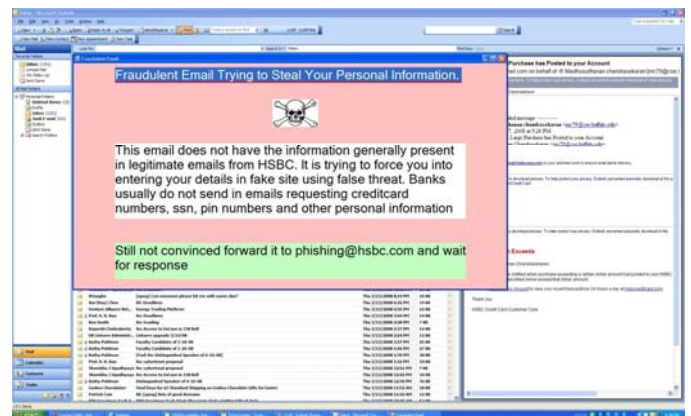


Fig. 3. Context-sensitive warning explaining the tone of the phishing email to the user

### A. Context Sensitive Warning Generation

The process of generating context-sensitive warnings is two-fold: (i) First, the suspect email’s content needs to be parsed and analyzed to extract underlying tone; (ii) then, the extracted tone must be communicated in an effective manner so that the user does not respond to the email. For the first step, during the training stage, each phishing email is broken down into a sequence of words  $W = \{w_1, w_2, \dots, w_n\}$ . The goal is then reduced to finding an optimal term set,  $W_{spoof} \subset W \times W$ , such that the “tone” conveyed in the spoofed email is accurately captured. At a broader level, even though  $W_{spoof}$  can be extracted using “bag-of-words” approaches that enumerate frequently appearing words in the email text, they, however, do not take into account the context (presence or absence) of a word with regard to other words in the text. As independent words (1-gram words such as account, credit card, user, name of the institution) that appear frequently in phishing emails also appear in emails sent from legitimate institutions, these 1-gram based approaches fail to scale well. In addition, these approaches do not account for grammatical relevance/context of the words appearing in the phishing email. In order to capture the tone conveyed in phishing emails, during the time of tokenization, we tag each word with its part-of-speech (POS), such as noun, verb, adjective, etc. We use Stanford log-linear part-of-speech tagger for this purpose, which is trained based on Penn Treebank English POS tag set [1]. As a result, each term in  $W_{spoof}$  contains a set of related words that appear in the phishing email along with their POS. Subsequently, insignificant words such as articles, conjunctions, prepositions and pronouns, which do not play any role in characterizing phishing emails are eliminated through a stop-word list. Once the insignificant words are removed, remaining words are normalized by converting them into their linguistic roots or “stems”. Stemming is a process in which morphological variants of words with similar semantic interpretation are transformed into their equivalent root. For example, in the context of phishing, the words submitting, submitted, submit appearing across different phishing emails are truncated into their root submit. The process of tagging the words with their POS is done prior to stemming to retain the context under which the words appear, as transforming the similar words into their root can alter the underlying POS. Porter’s algorithm, a popular stemming algorithm, is used for this purpose [13]. These transformed words are run through a standard thesaurus so that different stems with same meaning can also be normalized. Then the extracted words are passed through WordNet [7], an online resource where the nouns, verbs, adjectives and adverbs are grouped into a set of cognitive synonyms (synsets), which expresses a distinct concept. For example, the synsets for word immediate consists of words/phrases instantly, straightaway, straight off, directly, now, right away, at once, forthwith, like a shot. Each synset is also provided with a short summary (gloss), which provide more descriptive definitions or example sentences. The gloss for the synset of word immediate as

provided by WordNet is without hesitation or delay; with no time intervening. Even though the glosses provide description for synset, they do not annotate their underlying semantic roles, which is crucial in identifying the tone conveyed by the email. Using the lexical knowledge base FrameNet [2], synsets are mapped into one or more pre-annotated semantic frames depending on the type of event or state and the participants associated with them. These semantic frames are representations of situations involving various participants, properties, and other conceptual roles. Each frame has a set of associated words (lexical units), and is descriptive of a specific context/situation. Currently FrameNet consists of around 850 semantic frames with 135,000 annotated sentences. Also, each frame can be derived or related to many other frames. For the sake of illustration, the relationship between the *Taking\_Time* frame, which denotes time critical contexts and other similar frames in FrameNet is shown in Figure 4.

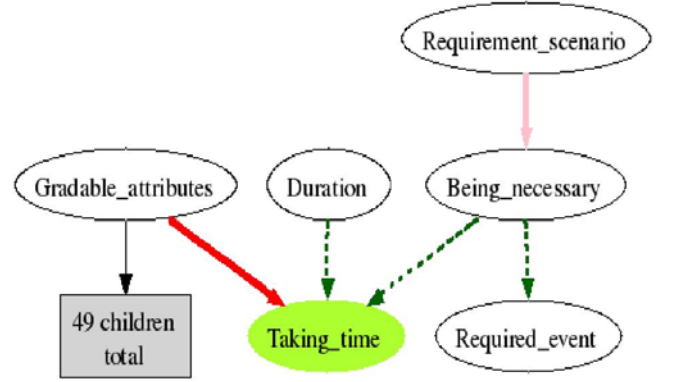


Fig 4. Relationship between the frame *Taking\_Time* and other abstract frames representing semantic roles that specify time boundedness in FrameNet

We adopt the algorithm given in [4] to convert the synsets into equivalent FrameNet frames. The algorithm, essentially, is a two step process: (i) In the first step, all candidate frames, whose lexical unit comprises of words in the synset or their variants (hypernyms/antonyms) are chosen. Then, for words that are not listed in the lexical unit of any frame, the names of the frames are checked to determine if they contain the words in synsets. For a frame to match there has to be at least 50% overlap between the word and the frame name. Finally, in order to select the best set of frames, all candidate frames are weighted depending on selection. The boost factor places more weight on the frames chosen for having the synset words in the lexical unit, as opposed to the frames which contains the words in their frame names. The weights for each frame are computed as shown in Algorithm 1.

```

input : WordNet Synsets for each word extracted from the email
output: A set of FrameNet frames with corresponding weights indicating the overall relevance
1 for each word  $w_s$  in the synset do
2   search_words = set of related hypernyms, antonyms corresponding to  $w_s$  from WordNet;
3 end
4 evoked_by_lexical_unit =  $\phi$ ;
5 evoked_by_name_match =  $\phi$ ;
6 for each frame  $f$  in FrameNet do
7   for each word  $w$  in search_words do
8     if  $w$  is in lexical unit of  $f$  then
9       evoked_by_lexical_unit( $f$ ) = evoked_by_lexical_unit( $f$ )  $\cup$   $w$ ;
10      spreading_factor( $w$ ) += 1;
11     else if ( $w$  has 50% match with  $f$ 's name) then
12       evoked_by_name_match( $f$ ) = evoked_by_name_match( $f$ )  $\cup$   $w$ ;
13       spreading_factor( $w$ ) += 1;
14     end
15   end
16 for each word  $w$  in search_words do
17   for each frame  $f$  in FrameNet do
18     Weight( $f$ ) =  $\sum_{w \in \text{evoked\_by\_lexical\_unit}(f)} \frac{\text{similarity}(w_s, w) * \text{boostfactor}}{\text{spreading\_factor}(w)}$ ;
19     Weight( $f$ ) +=  $\sum_{w \in \text{evoked\_by\_name\_match}(f)} \frac{\text{similarity}(w_s, w)}{\text{spreading\_factor}(w)}$ ;
20   end
21 end

```

Algorithm 1. Mining set of weighted FrameNet frames for WordNet synsets

In order to extract the tone of the email, a conglomerate set of frames is created by aggregating all the frames returned for each word in the phishing email message. In the training phase, we group these conglomerate sets into five categories, namely, justification, Penalty, Urgent Action, Reward and Concern, depending on the corresponding returned constituent frames and their weights. Also, appropriate context-sensitive warnings are assigned to the conglomerate sets, which describe the intent of the phisher to the recipients in an effective manner. In the testing phase, each email tagged as “suspicious” by CUSP is analyzed to see if it matches exactly/partially with one or more tagged conglomerate frames. Then depending on the match, the generated context-sensitive warnings are communicated to the user

## VI. EVALUATION

### A. Dataset

For our experiments, we consider a publicly available phishing corpus [11], which contains 434 phishing messages collected in a period of five months. Preprocessing is done to eliminate ill-formed emails that were not composed in English. Also, for the sake of brevity, messages with significant amount of spam (junk words) were discarded. The final list thus formed contained a total of 362 phishing emails. Almost all of the emails did not contain any (even random) user-specific data, barring a few exceptions. These set of emails were detected promptly by CUSP without any misclassification errors. A small fraction of emails (<2%) that impersonated eBay correctly had the user’s full name along with the user id. As these data can be obtained easily, it may be possible for an attacker to evade CUSP by using more focused attacks. Moreover, a few emails had fake transaction ids to fool the users into believing that they are sent by legitimate institution’s security department.

### B. Experiences with Context-Sensitive Warning Generation

The process of generating context-sensitive warning messages is two-fold: (i) In the first phase, we take a set of 200 email messages as training data and run it through the CUSP engine. The set of WordNet synsets returned are then passed into FrameNet to obtain the set of relevant frames. Each frame bears a part of the semantic structure of the tone conveyed in the email. For example, emails that impose a sense of urgency in the users (i.e., belonging to the Urgent Action category), have frames with names such as Response, Communication response, Requesting, Activity pause, Submitting documents compliance, etc. Similarly, emails that express security concern as an argument (i.e., belonging to Concern category) to deceive the users have frames with names such as Assistance, Personal relationship, Cause to start, Becoming aware, Evidence, Request, Education teaching, etc. Phishing emails that extort private data from users by threatening account revocation as a reason (i.e., frames falling into Penalty category), have frame names such as Inhibit movement, Thwarting, Compliance, Scrutiny, Attempt, Persuasion, Telling, etc. Similarly, the names of the frames corresponding to emails that give away incentives to users (i.e., frames belonging to the Reward category) for disclosing their account details are Telling, Personal relationship, Compatibility etc. Lastly, frames corresponding to Justification category include Waking up topic, Questioning, Leadership Request, Execute plan using, Protecting, etc. Then, we tag each conglomerate set (email) to only one of the five categories, even though they may have different overlapping individual frames. There were a total of 126 frames returned by FrameNet, which were formed by combining one or more of the 850 pre-annotated frames. Also, roughly on an average each email message in the training data returned a total of 15 different frames. The time taken to process each email message in the training phase is roughly in an order of few seconds; (ii) Once the training phase is completed, in the testing phase, each message is processed and depending upon the category they fall in appropriate context-sensitive message is conveyed to the user.

### C. Limitations of CUSP

There are three main limitations with CUSP: (i) First, as of now, the list of institutions that are vulnerable to phishing attacks is directly hard-coded in CUSP. Even though users are provided with an option to add their own custom tags, to make it more scalable, it is essential that these tags are managed remotely by a centralized system; (ii) Second, as CUSP operates only on text messages, it is still possible for a phisher to evade detection by encoding spoofed emails as images. To address such cases, we can provide warnings to users instructing them not to give away confidential information in response to such emails; (iii) Third, at this stage we only target phishing emails that are composed in English. However, as FrameNet like systems exist for other languages, porting CUSP to these languages is relatively easy.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we introduce CUSP, a customizable filter to separate phishing and legitimate emails based on user specific data contained in them. At the time of deployment, CUSP is bootstrapped with a list of known institutions that are vulnerable to phishing attacks. A user has the option of storing his personal information in the filter corresponding to subscribed institutions. Subsequently, every incoming email that purports to originate from the bootstrapped institutions is analyzed to see if it contains the corresponding user specific data. In case, if the data is absent, the email is tagged as phishing, and depending on the “tone” of the email, context-sensitive warnings are generated. We believe that these context-sensitive-warnings would help educate users about the potential hazards of phishing attacks. As a part of our future work, we would like to evaluate CUSP on emails sent over from legitimate organization and conduct a user study to test out the efficacy of context-sensitive warnings. Such a field test would help in fine tuning the working of CUSP so that better results could be achieved.

## REFERENCES

- [1] *Stanford log-linear part-of-speech tagger*. <http://nlp.stanford.edu/software/tagger.shtml>, March 2008.
- [2] C. F. Baker, C. J. Fillmore, and J. B. Lowe, “The Berkeley FrameNet project,” In *COLING/ACL-98*, pages 86–90, 1998.
- [3] J. C. Brustoloni and R. Villamarián-Salomoń, “Improving security decisions with polymorphic and audited dialogs,” in *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*, pages 76–85, New York, NY, USA, 2007. ACM.
- [4] Burchardt, K. Erk, and A. Frank, “A WordNet detour to FrameNet,” in *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen*, B. Fisseni, H.-C. Schmitz, B. Schrder, and P. Wagner, Eds, page 16, Frankfurt am Main, 2005. Lang, Peter.
- [5] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya, “Phishing email detection based on structural properties,” in *New York State Cyber Security Conference (NYS)*, Albany, NY, 2006.
- [6] N. Chou, R. Ledesma, Y. Teraguchi, and J. Mitchell, “Client-side defense against web-based identity theft,” in *11th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, 2004.
- [7] Cognitive Science Laboratory, Princeton University. *WordNet - a lexical database for the english language*. <http://WordNet.princeton.edu/>, 2008.
- [8] I. Fette, N. Sadeh, and A. Tomasic, “Learning to detect phishing emails,” in *16th international conference on World Wide Web (WWW)*, pages 649–656, Banff, Alberta, Canada, 2007. ACM Press.
- [9] Gartner Press Releases, “Gartner survey shows phishing attacks escalated in 2007; more than \$3 billion lost to these attacks.” <http://www.gartner.com/it/page.jsp?id=565125>.
- [10] M. Jakobsson, “The human factor in phishing,” in *Privacy & Security of Consumer Information*, 2007.
- [11] J. Nazario, *phishingcorpus homepage*, march 2008. <http://monkey.org/~jose/wiki/doku.php?id=PhishingCorpus>.
- [12] NetCraft, *Netcraft anti-phishing toolbar*, 2004.
- [13] M. F. Porter, “An algorithm for suffix stripping pages,” in *Readings in information retrieval*, pages 313–316, 1997.
- [14] Spooftick, *Spooftick toolbar*, 2004.
- [15] Y. Zhang, S. Egelman, L. Cranor, and J. Hong, “Phinding phish: An evaluation of anti-phishing toolbars,” in *Proceedings of Network & Distributed System Security Symposium (NDSS)*, 2007.
- [16] Y. Zhang, J. Hong, and L. Cranor. Cantina: a content-based approach to detecting phishing web sites. In *16th international conference on World Wide Web (WWW)*, pages 639–648, Banff, Alberta, Canada, 2007.
- [17] R. Dhamija, J.D. Tygar, and M. Hearst, “Cantina: a content-based approach to detecting phishing web site,” in *16th international conference on World Wide Web (WWW)*, pages 639–648, Banff, Alberta, Canada, 2007.

## Appendix: A

TABLE I  
SURVEY OF TOP 20 PHISHED BRANDS' SECURITY POLICY.

Name of the Bank	Token Identifier	Source	Website Link Specifying Privacy Policy
Amazon.com	Nil	S	<a href="http://www.amazon.com/gp/help/customer/display.html?nodeId=15835501">http://www.amazon.com/gp/help/customer/display.html?nodeId=15835501</a>
Bank of America Corporation	U	W	<a href="https://www.bankofamerica.com/privacy/Control.do?body=privacysecur_email_fraud">https://www.bankofamerica.com/privacy/Control.do?body=privacysecur_email_fraud</a>
Barclays Bank	FN, ADR	W	<a href="http://www.personal.barclays.co.uk/BRC1/jsp/brcontrol?task=homefreevi2&amp;value=9117&amp;target=_blank&amp;site=pfs">http://www.personal.barclays.co.uk/BRC1/jsp/brcontrol?task=homefreevi2&amp;value=9117&amp;target=_blank&amp;site=pfs</a>
Branch Banking and Trust Comp	FN, LF	W	<a href="http://www.bbt.com/bbt/about/privacyandsecurity/emailcommunication.html">http://www.bbt.com/bbt/about/privacyandsecurity/emailcommunication.html</a>
Capital One	FN, LF	S	<a href="http://capitalone.com/fraud/prevention/phishing.php?linkid=WWW\ Z\ Z\ Z\ FRD\ C1\ 01\ T\ FPRV1">http://capitalone.com/fraud/prevention/phishing.php?linkid=WWW\ Z\ Z\ Z\ FRD\ C1\ 01\ T\ FPRV1</a>
Citibank	FN, LF	S, W	<a href="https://www.citicards.com/cards/wv/detail.do?screenID=607">https://www.citicards.com/cards/wv/detail.do?screenID=607</a>
eBay	FN, UID	S, W	<a href="http://pages.ebay.com/education/spoofutorial/">http://pages.ebay.com/education/spoofutorial/</a>
Fifth Third Bank	FN	W	<a href="https://www.53.com/wps/portal/privacy/?New\ WCM\ Context=/wps/wcm/connect/FifthThirdSite/Global+Utilities/Privacy(%20)%26(%20)Security/#">https://www.53.com/wps/portal/privacy/?New\ WCM\ Context=/wps/wcm/connect/FifthThirdSite/Global+Utilities/Privacy(%20)%26(%20)Security/#</a>
HSBC Bank	FN, LF	S, W	<a href="http://www.us.hsbc.com/1/2/3/personal/inside/securitysite/your-responsibility">http://www.us.hsbc.com/1/2/3/personal/inside/securitysite/your-responsibility</a>
HSBC Credit Card	FN, LF	S, W	<a href="http://www.us.hsbc.com/1/2/3/personal/inside/securitysite/your-responsibility">http://www.us.hsbc.com/1/2/3/personal/inside/securitysite/your-responsibility</a>
JP Morgan Chase and Co	FN, LF	S, W	<a href="http://www.chase.com/ccp/index.jsp?pg_name=ccpmapp/shared/assets/page/Report_Fraud#5">http://www.chase.com/ccp/index.jsp?pg_name=ccpmapp/shared/assets/page/Report_Fraud#5</a>
National City	U	W	<a href="http://www.nationalcity.com/about/privacy/identity/default.asp">http://www.nationalcity.com/about/privacy/identity/default.asp</a>
PayPal	FN	W	<a href="http://www.paypal.com/cgi-bin/webscr?cmd=p/gen/fraud-prevention-outside">http://www.paypal.com/cgi-bin/webscr?cmd=p/gen/fraud-prevention-outside</a>
Poste Italine	U	W	<a href="http://www.poste.it/online/phishing.shtm">http://www.poste.it/online/phishing.shtm</a>
Regions Bank	U	R	<a href="http://www.regions.com/about_regions/email_fraud.rf">http://www.regions.com/about_regions/email_fraud.rf</a>
US Bank	FN, LF	W	<a href="https://www4.usbank.com/internetBanking/en_us/info/BrowserRequirementsOut.jsp">https://www4.usbank.com/internetBanking/en_us/info/BrowserRequirementsOut.jsp</a>
Volksbanken Raiffeisenbankeni	U	W	<a href="http://www.vr-networld.de/c132/default.html">http://www.vr-networld.de/c132/default.html</a>
Wachovia	FN, LF	R	<a href="http://www.wachovia.com/securityplus/page/0,,10957\ 10970,00.html">http://www.wachovia.com/securityplus/page/0,,10957\ 10970,00.html</a>
Wells Fargo	U	W	<a href="https://www.wellsfargo.com/privacy_security/fraud/report/fraud?requestid=394409">https://www.wellsfargo.com/privacy_security/fraud/report/fraud?requestid=394409</a>
Western Union	U	W	<a href="http://www.westernunion.com/info/fraudProtectYourself.asp">http://www.westernunion.com/info/fraudProtectYourself.asp</a>

- **All the companies indicate that they do not send emails requesting confidential information.**
- Token identifiers indicate what user specific data is included in the companies' email to the customers
  - FN - Full Name, UID - Username, LF - Last four digits of Account Number, NA - No private data, U - Unverified/Not known
- Source indicates where the information about company's security policy was obtained (W - Website, S - Sample email.)