Identification and Detection of Phishing Emails Using Natural Language Processing Techniques

Shivam Aggarwal
BT Kumaon Institute of Technology
Dwarahat, INDIA
shivam221292@ieee.org

Vishal Kumar
BT Kumaon Institute of Technology
Dwarahat, INDIA
vishalkumar@kecua.ac.in

S D Sudarsan
ABB Corporate Research
Bangalore, INDIA
sudarsan.sd@in.abb.com

ABSTRACT
Phishing refers to fraudulent social engineering techniques used to elicit sensitive information from unsuspecting victims. In this paper, our scheme is aimed at detecting phishing mails which do not contain any links but bank on the victim's curiosity by luring them into replying with sensitive information. We exploit the common features among all such phishing emails such as non-mentioning of the victim's name in the email, a mention of monetary incentive and a sentence inducing the recipient to reply. This textual analysis can be further combined with header analysis of the email so that a final combined evaluation on the basis of both these scores can be done. We have shown that this method is far better than the existing Phishing Email Detection techniques as this covers emails without links while the pre-existing methods were based on the presumption of link(s).

Categories and Subject Descriptors
K.4 [**Computers and Society**]: Electronic Commerce; K.4.4 [**Electronic Commerce**]; [Security]

General Terms
Verification, Security, email.

Keywords
Privacy, Phishing, Verification.

1. INTRODUCTION
Phishing emails[1] have been a major concern for the web users worldwide. According to Kaspersky Lab report [21], 37.3 million users experienced phishing attacks in 2013 and are increasing in numbers year by year. FraudWatch International, a privately owned Internet Security company lists the recent phishing attacks [22]. Phishing email aims to elicit sensitive information from the user which, on being stolen, can cause harm (while the attacker may get unsolicited benefits). The information may be asked for either directly or indirectly, and in various ways that can fool the victim. There isn't a complete solution till now for this problem primarily because of there being a number of ways of phishing which cannot be detected by machine and can force/lure/tempt a human to reveal sensitive information. Most times, the victim gives away information thinking that it's going into safe or trustworthy hands (like when giving away passwords/PIN's, etc.), while at other times the user doesn't realize what harm can be done by revealing the information being asked for(like while revealing personal details like DOB, Address, etc.).

Attacks are typically carried out via e-communication channels such as email or instant messaging by attackers posing as legitimate and trustworthy entities. These attackers pose as some trustworthy entity and try to make the victim voluntarily give in his/her sensitive details which may be personal or belonging to an organization. These sensitive details can be usernames and passwords of online banking accounts or can even be information used to create fake Identities of the victim. Such information can lead to both monetary and identity thefts.

In this paper, we focus on email communication, which is the most popular medium to launch such attacks [1]. Our scheme is aimed at detecting phishing mails which do not contain any links but bank on the victim's curiosity by luring them into replying with sensitive information. We aim to exploit the common features among such phishing emails such as no mention of the victim's name in the email, a mention of some amount of money and a sentence asking for a reply. The victim's name is easily available to the email service provider; the mention of money is determined by finding all appearances of a currency symbol or currency name in the email; the mention of a reply inducing sentence can be found out by searching for words asking for reply which are present in the sample mails.

Our proposed method makes use of Natural Language Processing (NLP) and WordNet[2]. NLP is an area concerned with enabling the computer to derive meaning from the language spoken by humans (Hindi, English, Bengali, French, etc.). WordNet is a database that combines the work of a dictionary and a thesaurus.

Natural Language Processing
NLP is an area of Computer Science concerned with enabling the computer to derive meaning from the language spoken by humans. The idea is to make a computer similar to human in a way that it can process data and instructions in Natural Language in a meaningful manner. Out of the many techniques used in the

[1] We use "email" and "mail" interchangeably in this paper.

[2] http://wordnet.princeton.edu

SIN '14, September 09 - 11 2014, Glasgow, Scotland Uk
Copyright 2014 ACM 978-1-4503-3033-6/14/09…$15.00.
http://dx.doi.org/10.1145/2659651.2659691

field of NLP, the two used here are Part of Speech (PoS) Tagging and Word Stemming.

PoS Tagging is concerned with tagging each word in the text with the correct PoS in the given context. For example, the word *plant* may be used as a verb in one case ("We should *plant* more and more trees") whereas it may refer to a manufacturing unit in another case ("The construction of the steel *plant* has created job opportunities for the local lads"). The context of the word decides which PoS tag must be associated with it. There are various methods used to handle the challenges of PoS Tagging. The PoS tags used in the field are not confined to the PoS of English Language. There are many more tags than specified by English language.

Morphology is used for conversion of English words to base form. In morphology, plurals, verb endings and pronoun case are taken care of and words in these forms are converted to their base forms (e.g. "The profit will be shared between me and you" will be converted to "The profit will be share between I and you"). Things like comparative adjectives, derived nominal, etc. are not considered here. We use the API Stanford CoreNLP [20] for NLP techniques in our implementation. In Stanford CoreNLP, morphology is based on finite-state transducer [23].

WordNet

WordNet is a database that combines the functionalities of a dictionary and a thesaurus. It contains the synonyms as well as the meanings of a word. The words in WordNet are arranged hierarchically. The basic building block is a synset which is the set of all words which represent the same concept. For example the words *answer*, *reply* and *response* refer to the same concept which is *a statement (either spoken or written) that is made in reply to a question or request or criticism or accusation*[3]. So these words will be a part of the same synset, the PoS being noun in this case. The relation between synonyms of a synset is called a synonymy relation. Other relations are hyponymy and hypernymy relations among words in the WordNet tree. Hyponymy helps to group words hierarchically and is a relation of sub-ordination. For example, *car* is a hyponym of word *vehicle*, and so *vehicle* is the hypernym of the word *car*. As we follow the hyponym links in the WordNet hierarchy, we deviate from the actual concept represented by the original word. This is an important point to consider in our scheme, which is discussed later.

2. CHARACTERISTICS OF PHISHING ATTACK

To understand the nature of phishing emails, that are link-less, we collected and analyzed about 600 phishing emails collected over a period of 6 months. We identified and selected certain common factors, which are present in link-less phishing emails but not in other mails. These factors are lack of the recipient's name, mention of money, reply inducing sentence and a sense of urgency. Another factor is that the sender's email address shown in the header is different from the reply-to email id. This factor is covered in header analysis by the authors of [18].

We present our strategy to detect the above mentioned factors in a mail and the way we combine these factors to get a final score is discussed later in Section 4.

[3]http://www.thefreedictionary.com/statement

2.1 Absence of recipient's name

The people who send these emails which do not contain links but depend on the victim's reply for carrying out the attack generally find the email lists from websites or use web crawlers. This specific type of email is always seen to be coming without the name of the recipient mentioned anywhere in the email. It generally starts with phrases like "Dear Beloved", "Dear son of God", "Dear Friend", "Hi Dear", "My Dear Beneficiary" and various such phrases which lack the recipient's name. We find this to be an important factor in detecting such a mail.

2.2 The mention of money

The easiest way a stranger can get someone to reply to their emails seems to be promising a good amount of money. The lust for money, particularly "free money" with no strings attached is exploited by most attackers. They promise the victim a sum of money in some way or the other so that the victim is lured into replying to them. Once the victim starts to believe that he/she is going to get the promised amount of money and that the people making the promise are authentic, then the attackers ask them for sensitive information or ask them to transfer a sum of money to an account and then they disappear.

2.3 Reply inducing sentence

The email generally closes with a sentence which asks the user to reply to a particular email address. If anyone does reply, the actual mind games start. The attackers pose as the entity they mentioned in the email, and then do everything they can to lure the victim into confidence so that he/she may reply and provide with sensitive details.

2.4 Sense of urgency

The reply luring sentence generally implies a sense of urgency so that the victim replies as soon as possible. This perhaps, serves two purposes. First, the victim has less time to think logically, since once replied the attacker has at least some information which otherwise was not available. Second, the chance of the attacker's email reported and subsequently blacklisted or blocked for communications is reduced. This happens, as emails once replied are considered as non-spam and/or malicious.

These three characteristics seem significant and sufficient to detect the phishing mails. What is important to note is that it is valid in combination. For example, some legitimate email may be related to money and with urgency, but will have the name and associated information of the recipient mentioned.

If one reads the emails sent by such attackers, one may find that there is a certain pattern to the way they tell you about things. Such attackers are generally scammers. These scammers usually contact potential victims by email or letter and offer a share in a large sum of money that they want to transfer out of their country. They may tell about money purportedly trapped in central banks during civil wars, often in countries currently in the news. Or they may tell about purported massive inheritances they are "entitled to" but are difficult to access because of government restrictions or taxes in their country.

Scammers ask victims to pay money or give them bank account details to help them transfer the money. Victims are then asked to pay fees, charges or taxes to help release or transfer the money out of the country through victim's bank. These 'payments' may even start out as quite small amounts. If paid, the scammer brings up requirements of new 'fees' that require payment before victim can receive the 'reward'. They will keep making up these false

requirements until they think they have got all the money they can out of the victim. Victim, of course, will never be sent the money that was promised.

In the 600 emails collected by us we found that the excuses made up by attackers to make the victim believe that the latter will get a lot of money are various and unrelated to each other. The excuses range from tragedy of a businessman's wife left with huge amount of money after his death, to the businessman himself offering the victim a business opportunity to work with him, albeit in an illegal and immoral way.

3. RELATED WORK

Phishing has been studied from various perspectives by many researchers and scholars. Different people have studied from different perspectives and devised many server-side and browser-side strategies, strategies for education/training of prospective victims and evaluation of anti-phishing tools has been performed. Then there have been studies to find the reasons behind success of phishing attacks. We try to provide a brief description of all such previous studies here.

3.1 Email and Web Page based phishing detection schemes

The two broad categories of phishing detection schemes are web page based and email content based. The former depends on the content of the web page which is opened by clicking a link, while the latter is concerned about the content of the email received by the victim.

The two categories can be further classified into 3 sub-categories:

1. Schemes based on information retrieval

2. Machine learning based techniques

3. String, pattern and visual matching based detection schemes.

Initially the scheme used for protection against phishing was the use of blacklists (list of reported websites). The links were blocked by browsers if found in the blacklist. Ludl et al. [3] performed an analysis of such techniques to test their effectiveness and found that these are effective and useful solutions. But the major drawback is that these can't protect against zero-day phishing attacks i.e. a link used for phishing which is not already in the blacklist.

3.2 Phishing Detection over Web page Content

Cantina [5, 6] is a scheme using information retrieval and text mining algorithms to analyze the content of web pages for detection of phishing. In [7], a research team from Google has presented a technique that analyzes both URL and the web page content to identify phishing webpages. This is a machine learning based technique achieving an accuracy of 90%.

3.3 Phishing Detection Using URL analysis

Methods, which are completely URL based, have been proposed as they try to analyze only the structure of a URL in order to classify it as benign or malicious. These methods do not consider the content of the target webpage. The authors of [8] have described many features to differentiate between the bad and good URLs and use these features to develop a logistic regression filter which has a high accuracy. LinkGuard [9] tries to extract the common characteristics of URLs embedded in mails provided by Anti Phishing Work Group, successfully detecting 195 out of 203 phishing attacks.

3.4 Phishing Detection over Email Content

The schemes based on email content mainly use machine learning techniques on a set of features extracted from emails identified as phish. The selected features are such that they help differentiate the phishing emails from benign ones. Statistical classifiers are trained on such sets of features, after which they are employed to detect phishing emails in the received emails. The number and types of features used by these methods is where they differ from one another. The main problem associated with such classifiers is that they need to be updated regularly by training in new set of data as and when it becomes available. Such schemes have been presented in [10-15]. A comparison of such methods has been presented in [16].In [17], a method based on header, link and text analysis has been proposed which is heuristics based. A classification in flash and non-flash attacks has been given in [2].

3.5 Phishing Detection over Email Content using NLP

Verma *et al* [18] are probably the only scholars to have previously used NLP Techniques for detecting phishing emails. They have shown their scheme to be better than all existing schemes. But they have a link-oriented scheme. If there is no link present in the email, then it is always going to pass through their PhishNet-NLP as an innocuous email.

3.6 Training the user herself to identify phishing attack

The authors in [25] mention common characteristics of phishing emails but haven't given any way to detect them. For example, they have said that a PROMISING (OR LUCRATIVE) OFFER present in a mail increases its chances of being a phish. But the authors do not provide a way to differentiate between a lucrative and non-lucrative offer. Absence of name of recipient, asking for a reply, sense of urgency, and some characteristics related to embedded links are also given to identify phishing emails. But in our paper we are showing a way to identify those characteristics in mail via a list of easy to check questions.

In this paper, we focus on phishing emails that have no link or URL.

4. THE PROPOSED SCHEME

This section explains the methodology used in the implementation of this strategy.

4.1 Detecting the absence of names

Detecting the absence of names of the recipient is a trivial task if we have the name of the individual who holds the account. But there's a problem of spelling mistakes including deliberate spelling mistakes. To deal with this, the following method can be used.

Let **F**, **M** and **L** denote sets of common spelling variants (as entered by the user) of the user's first name, middle name (if any) and last name respectively. Now the set **N** is calculated to find all possible name variants of the user by finding union of cross products of sets **F**, **M**, and **L** taking one of them at a time, two of them at a time and all three at a time in all possible permutations:

$$\boldsymbol{N = F \cup M \cup L \cup FM \cup FL \cup MF \cup ML \cup LF \cup LM \cup FLM \cup FML \cup MLF \cup MFL \cup LMF \cup LFM}$$

Now that we have all the possible name variants of the user, we can find whether her name is present in the mail or not.

4.2 Detecting mention of money

If money is mentioned in the email, then it would be present in the form of an amount followed or preceded by a currency symbol or name.

The presence of the currency symbol or name in the email can be checked by keeping a set of names and symbols of all the currencies in the world and their common variants. (By common variants we mean the way we call the currency, e.g. INR or Rs. for Indian Rupee, USD or $ for United States Dollar, etc.) We found that mostly popular international currencies are used. Even if less popular currencies are used, an equivalent popular currency is also mentioned.

For testing purposes, and on the basis of emails studied by us, only the currencies of India, Great Britain and USA have been considered in the implementation of this method by us.

4.3 Detecting presence of reply inducing sentence

A reply inducing sentence will generally contain a word or phrase which asks the user to reply to the email. On the basis of study of such mails over a six month period, the words that have been found to ask for a reply are write, contact, get back, reply, response, forward, send and hear. Let set R be a set of all such words which ask for a reply from the users.

For any set X containing words along with a sense for each word, let Synset(X) = {synset(x) | x ε X}, where synset(x) is the WordNet synset of x for the specified sense. For natural number i ≥ 1, let $Hypo^i$(Synset(R)) denote the union of all the synsets reached by following up to i hyponymy links from the synsets in Synset(R). We let SR = $Hypo^4$ (Synset(R)) be the set of special verbs.

We need to find if any of these words are present in the email text. For that, the email text file needs to be stemmed to base form of all the words that are present in the file. This has been done via Stanford CoreNLP API which is used to first tag the file with the appropriate POS Tags, and this tagged file is later used as input to the Morphology object which stems the words of the tagged file to their base form.

4.4 Detecting sense of urgency

Let U = {now, today, instantly, straightaway, directly, once, urgently, desperately, immediately, soon, shortly, quickly}. These are words that induce a sense of urgency.

If a sentence that contains the word from SR also contains a word from U, then it means that the attacker has asked the user to reply with a sense of urgency associated with the request. This is important to take into consideration as the final score needs to be substantiated using this factor.

For any word r ε SR, let

$$\text{Score(r)} = \frac{n \times (m+s+u)}{2^L}$$

where:

n = 1 if there is no mention of name in the email, otherwise 0.

m = 1 if there is a mention of money in the email, otherwise 0.

s = 1 if there is a word from SR in the email, otherwise 0.

u = 1 if there is a mention of urgency in the same sentence as there is a word from SR

L = number of Hyponym links followed to reach the word r from the words in R.

The final score of the email is the maximum score obtained among all r ε SR.

Combining the score with Header Analysis

Performing Header Analysis has been left as a future work. Moreover, there is a comprehensive header analysis in PhishNetNLP [18]. This score obtained in this equation can be used in place of TextScore in the work of Verma et al [18]. Their Header Analysis can combine with this score in place of their TextScore can help detect a wider range of phishing emails.

5. RESULTS

We tested 600 phishing emails and 400 innocuous emails using this process. To measure the quality of phishing detection schemes, we need to consider the number of True Positives, False Positives, True Negatives and False Negatives marked by our implementation. These are explained below:

i. True Positives – This is the number of real phishing emails that have been marked positively by our implementation.
ii. False Positives – This is the number of non-phishing emails that have been marked as phishing emails by our implementation.
iii. True Negatives – This is the number of non-phishing emails that have been marked negatively by our implementation.
iv. False Negatives – This is the number of phishing emails that have been marked as non-phishing by our system.

In our case, we took 1000 emails and classified them as phishing or non-phishing manually. Then our implementation was made to process each mail one by one and the result for each mail was compared with our classification of that mail as phish or non-phish. The details are as follows:

Number of True Positives (TP) = 596
Number of False Positives (FP) = 2
Number of True Negatives (TN) = 398
Number of False Negatives (FN) = 4

Precision = $\frac{TP}{TP+FP} = \frac{596}{598} = 0.996$

Recall Sensitivity = $\frac{TP}{TP+FN} = \frac{596}{600} = 0.993$

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN} = \frac{994}{1000} = 0.994$

The false positives were found to be emails from known senders, in which the salutation without name (e.g. “Dear Sir” is used by juniors for seniors) was used and money was mentioned. False negatives were emails from unknown senders which had single sentences without a mention of money (in form of currency) and without a request for reply. These single sentence emails contain innocuous looking sentences such as those asking about the health of the recipient or just a simple “How are you”. Once the recipient replies to such a sentence (out of curiosity to know about who the sender is), the actual attack is carried out. This becomes a

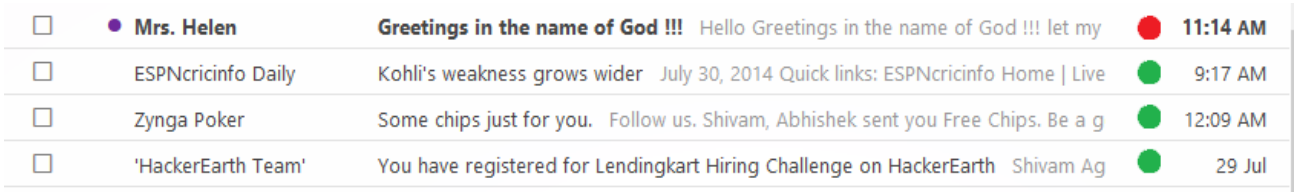

Fig. 1 – Showing the Danger Sign for Emails Detected as Phish

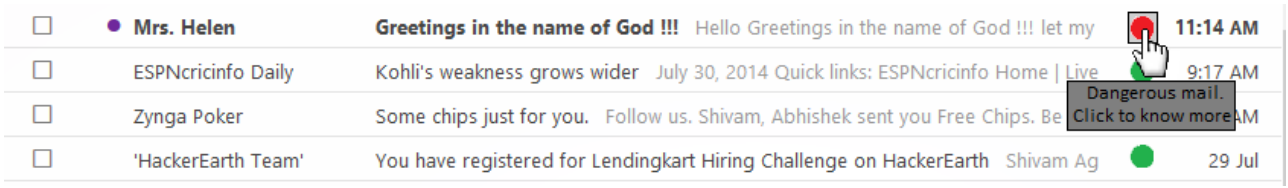

Fig. 2 – Hovering over the Danger Sign Shows "The Prompt" to Click to "know more"

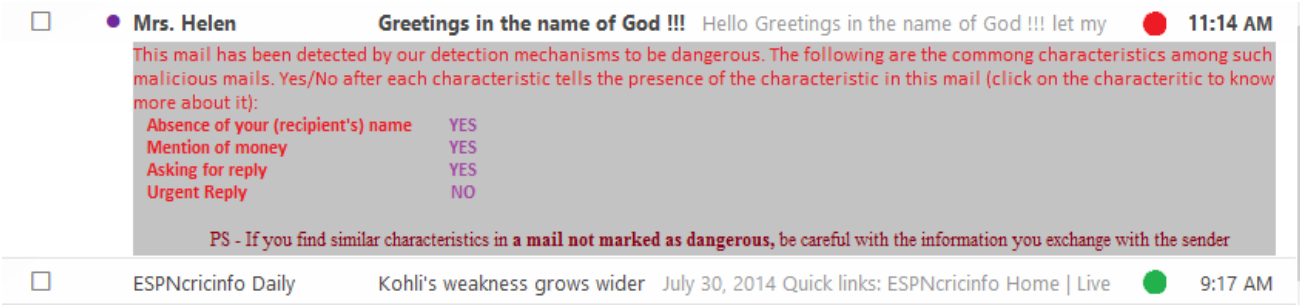

Fig. 3 – Clicking the Danger Sign will Show More Details about the Mail

conversation thread which is not checked subsequently, as the sender is inferred by the existing mechanisms to be known to the recipient.

This serves as a motivation to include header analysis of the email in combination with our scheme, in order to improve the performance by zeroing out the FP and FN counts.

5.1 Letting it Known to the End User

The main problem with people getting trapped day after day is that they don't get to know the exact reasons why a mail was sent to their spam box. No matter what techniques are used by the anti-phishing techniques, attackers find a way around it to trap people. Letting the prospective victim know about the characteristics found in Phishing emails can help a long way in avoiding the attacks which bypass the filters. With this in mind we devised a way to let the end user know on what basis their mails were marked as phish. In this the email client (software) will get the required information about why the email was marked as phish.

It shall display a danger sign, as shown in the Fig.1, in front of emails we marked using our method. If the user clicks that sign, she'll be shown more information, as shown in Fig. 2 and Fig. 3. Clicking on the danger sign to will show the user the following details about the mail:

Absence of name? - Yes/No

Mention of money? -Yes/No

Asking for reply? - Yes/No

Urgent Reply? - Yes/No

Then a note, "If you find similar characteristics in a mail not marked as dangerous, be careful with the information you exchange with the sender" will be displayed. Clicking on any of the four factors in the list shall give detailed information about each factor, as mentioned in Section 2.

We can see that this method provides fairly high levels of statistical measures for the quality of Phishing Email detection mechanisms. When this is combined with Header Analysis, the results can be further improved. Moreover, we are yet to implement the calculation of the value of L in our equation using WordNet, which is expected to further improve our scheme. The measure of L will help us decide an optimum threshold score to mark a mail as a phish or benign.

6. CONCLUSION AND FUTURE WORK

In this work, we used NLP techniques to detect phishing emails without links. The accuracy of the score of the email can be further improved by calculating L which is the number of Hyponym links that shall be followed in order to reach the given reply inducing word present in the email, from the set of words R (which is the set of synsets of all the words that induce a reply and have been found in the sample phishing mails). The use of Header Analysis has not been done in our work as it has already been done in [19]. The idea was to calculate the text score of mails without links and combine it with the available heuristics (link score, header score and context analysis) present in [19].

Currently, L is set to 1 by default. This means that the score will always be one out of 0.5, 1.0 and 1.5. As L increases, the score will decrease exponentially. This is justified as the meaning of word drifts away from the actual meaning as we follow

hyponymy links. This factor L accounts for the imprecision of words as we follow the hyponymy links and helps to reduce the score accordingly. This way (intuitively) only a score less than 0.25 would mean a harmless email, when the formation of the set **SR** and the subsequent calculation of L are integrated into this work. We plan to use RiTa WordNet API [24] to access the WordNet database programmatically.

There is still a room for improvement in the proposed scheme. One latest threat is that the attackers are now sending the intended text of the email in attachments. This method needs to be implemented in such a way that it can extract text from both the text part of the email and also the attachment. The difficult part here is that the attachment may not be a text file, but instead it may be an image file with perceivable text in it. The scheme would then require Optical Character Recognition techniques to detect the text content in each image and then use that information to perform text-based tests for phishing detection.

It is a saying in the world of Information Technology that the developers rectify their mistakes only after the hackers find them. In the case of phishing mails too, this saying holds true. Our method depends on the types of mails we have seen till now. There may be more types of attacks invented by the attackers to bypass the checks imposed by this scheme. That is beyond control.

ACKNOWLEDGMENTS

We are grateful to Dr. Rakesh Verma, Department of Computing, University of Houston for providing us with guidance with regard to using Natural Language Processing techniques to detect phishing emails. His work [18] served as the inspiration for us to devise a scheme to detect phishing emails using NLP techniques.

7. REFERENCES

1. Parno, B., Kuo, C., Perrig, A.: Phoolproof Phishing Prevention. In: Di Crescenzo,G., Rubin, A. (eds.) FC 2006. LNCS, vol. 4107, pp. 1–19. Springer, Heidelberg (2006)

2. Irani, D., Webb, S., Giffin, J., Pu, C.: Evolutionary study of phishing. In: 3rd Anti-Phishing Working Group eCrime Researchers Summit (2008)

3. Ludl, C., McAllister, S., Kirda, E., Kruegel, C.: On the Effectiveness of Techniques to Detect Phishing Sites. In: H¨ammerli, B.M., Sommer, R. (eds.) DIMVA 2007.LNCS, vol. 4579, pp. 20–39. Springer, Heidelberg (2007)

4. Sheng, S.,Wardman, B.,Warner, G., Cranor, L., Hong, J., Zhang, C.: An empirical analysis of phishing blacklists. In: Proc. 6th Conf. on Email and Anti-Spam (2009)

5. Zhang, Y., Hong, J., Cranor, L.: Cantina: a content-based approach to detecting phishing web sites. In: Proc. 16th Int'l Conf. on World Wide Web, pp. 639–648.ACM (2007)

6. Xiang, G., Hong, J., Rose, C.P., Cranor, L.: Cantina+: A feature-rich machine learning framework for detecting phishing web sites. CM Trans. Inf. Syst. Secur. 14,21:1–21:28 (2011)

7. Whittaker, C., Ryner, B., Nazif, M.: Large-scale automatic classification of phishing pages. In: Proc. of 17th NDSS (2010)

8. Garera, S., Provos, N., Chew, M., Rubin, A.: A framework for detection and measurement of phishing attacks. In: Proc. 2007 ACM Workshop on Recurring Malcode, pp. 1–8 (2007)

9. Chen, J., Guo, C.: Online detection and prevention of phishing attacks. In: First Int'l Conf. on Communications and Networking in China, ChinaCom 2006, pp. 1–7. IEEE (2006)

10. Fette, I., Sadeh, N., Tomasic, A.: Learning to detect phishing emails. In: Proc. 16th Int'l Conf. on World Wide Web, pp. 649–656. ACM (2007)

11. Chandrasekaran, M., Narayanan, K., Upadhyaya, S.: Phishing email detection based on structural properties. In: NYS CyberSecurity Conf. (2006)

12. Bergholz, A., Chang, J., Paaß, G., Reichartz, F., Strobel, S.: Improved phishing detection using model-based features. In: Proc. Conf. on Email and Anti-Spam, CEAS (2008)

13. Basnet, R., Mukkamala, S., Sung, A.: Detection of phishing attacks: A machine learning approach. In: Soft Computing Applications in Industry, pp. 373–383 (2008)

14. Bergholz, A., Beer, J.D., Glahn, S., Moens, M.F., Paaß, G., Strobel, S.: New filtering approaches for phishing email. Journal of Computer Security 18(1), 7–35 (2010)

15. Gansterer, W.N., P¨olz, D.: E-Mail Classification for Phishing Defense. In: Boughanem, M., Berrut, C., Mothe, J., Soule-Dupuy, C. (eds.) ECIR 2009. LNCS, vol. 5478, pp. 449–460. Springer, Heidelberg (2009)

16. Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: A comparison of machine learning techniques for phishing detection. In: Proc. Anti-phishing Working Group's 2nd Annual eCrime Researchers Summit, pp. 60–69. ACM (2007)

17. Yu, W., Nargundkar, S., Tiruthani, N.: Phishcatch-a phishing detection tool. In: 33rd IEEE Int'l Computer Software and Applications Conf., pp. 451–456 (2009)

18. Verma et al: Detecting Phishing Emails the Natural Language Way. In: ESORICS 2012, LNCS 7459, pp. 824–841, 2012.

19. Fellbaum, C. (ed.): WordNet An Electronic Lexical Database. MIT Press (1998)

20.The Stanford Natural Language Processing Group. (2014, March). Stanford CoreNLP 3.3.0. Available: nlp.stanford.edu/software/corenlp.shtml

21. Kaspersky Lab report: More that 37.3 million users reported phishing last year. Available: http://www.kaspersky.com/about/news/press/2013/Kaspersky_Lab_report_37_3_million_users_experienced_phishing_attacks_in_the_last_year

22. Fraudwatch International Phishing Alerts. Available: http://www.fraudwatchinternational.com/phishing-alerts

23. Morphological and Orthographic Tools for English. Available: http://www.sussex.ac.uk/Users/johnca/morph.html

24. RiTa WordNet – Rednoise. Available: http://www.rednoise.org/rita/wordnet/documentation

25. Lotter A., Futcher L.: A Framework to assist Email users in the identification of Phishing Mails. In: Proc. 8th Int'l Symposium on Human Aspects of Information Security and Assurance (2014).