MJRET

OPEN ACCESS

Multidisciplinary Journal of Research in Engineering and Technology

www.mjret.in

eISSN:2348-6953

Identification and Detection of Phishing Email using Natural Language Processing Techniques

Ms. Smita Deepak Kanerkar, Prof. Flavia Gonsalves

Masters of Computer Applications, MET Institute of Computer Science,
Bandra West, Mumbai, India

Abstract: *Email is still the most commonly used medium to launch phishing attacks. This scheme utilizes all the information present in an email, namely, the header, the links and the text in the body. Although it is obvious that a phishing email is designed to elicit an action from the intended victim, none of the existing detection schemes use this fact to identify phishing emails. This detection protocol is designed specifically to distinguish between "actionable" and "informational" emails. To this end, we incorporate natural language techniques in phishing detection. We also utilize contextual information, when available, to detect phishing: we study the problem of phishing detection within the contextual confines of the user's email box and demonstrate that context plays an important role in detection. This is the first scheme that utilizes natural language techniques and contextual information to detect phishing. This protocol detects phishing at the email level rather than detecting masqueraded websites. This is crucial to prevent the victim from clicking any harmful links in the email. This implementation is called PhishNet-NLP, operates between a user's mail transfer agent (MTA) and mail user agent (MUA) and processes each arriving email for phishing attacks even before reaching the inbox. 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. 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).*

Keywords*: Phising Detection Techniques, ,Attacks, Phisnet-NLP, Stopwords, Stemming,Context Score, Tokenization ,Data Extraction, Normalization*

1. INTRODUCTION

Phishing is a social engineering threat aimed at gleaning sensitive information from unsuspecting victims. Attacks are typically carried out via communication channels such as email or instant messaging by attackers masquerading as legitimate and trustworthy entities. In this paper, we focus only on email communication as it is the most popular medium to launch such attacks. Detecting phishing email messages automatically is a non-trivial task.Primary contribution in this paper is a comprehensive and effective natural language based phishing detection scheme. Our scheme uses the information present in the email header, text in the email body and the links embedded in the email. We make use of novel techniques to process the header and link information, and deeper natural language techniques to process the text information. To the best of our knowledge, this is the first natural language based scheme for phishing detection.

Natural language processing (NLP) by computers is well-recognized to be a very challenging task because of the inherent ambiguity and rich structure of natural languages. Perhaps this explains why previous researchers have not used NLP techniques for email phishing detection. Despite this difficulty, here we show that our scheme outperforms all existing phishing detection strategies in the literature and obtains a phishing detection rate of 97% or better with very low false positives (0.7-0.8%). This scheme is built on the observation that the fundamental difference between a phishing and a legitimate email lies in its objective. While a legitimate email typically conveys some information to the reader, a phishing email is designed to elicit a response. This response often involves making the reader click a link with the intention of obtaining personal sensitive information. None of the detection schemes in the literature make use of this distinction to detect phishing emails. Focus is on objectives that are typical of phishing emails - language that intends to create a sense of urgency, threat, worry, concern or offers an incentive to the user to perform an action.

This implementation PhishNet-NLP operates between a user's MTA and MUA and processes each arriving email for phishing attacks. This prevents the user from clicking any harmful link in the email. This approach is in contrast to schemes that analyze the target websites for authenticity. The motivation to operate at the email level is due to the fact that clicking on the link and visiting a phishing website exposes the user to potential malware that could be installed by the website. Furthermore, it is our objective to maximize the distance between the user and the phisher - clicking a malicious link puts the user closer to the threat. The added advantage of this approach is that ISPs and email providers may now be able to prevent such emails from being delivered to the user thereby saving precious bandwidth as well.

2. PRIOR WORK

Phishing is primarily a social engineering attack and has attracted a lot of research interest in this context. Different research groups have studied this problem from various perspectives: server-side and browser-side strategies, education/training, evaluation of anti-phishing tools, detection schemes and finally studies that analyze the reasons behind the success of phishing attacks. Briefly outlined the prior related work on phishing categorized by research objectives.

Phishing Detection Schemes - Email and Web pages -There are two primary classificationsof phishing detection schemes: schemes that detect phishing based on analyzing the content of the target web pages (targets of the embedded email links) and schemes that operate directly on the content of the emails. Before such schemes, the most popular was the integration of blacklist-based anti-phishing techniques into browsers used to test the effectiveness of the blacklists maintained by Google and Microsoft to understand the viability of this approach. It is also been shown that blacklists are ineffective for protecting users from phishing attacks initially and that their effectiveness increases with time.

Phishing Detection Over Email Content - Uses machine learning techniques on a feature set designed to highlight user-targeted deception in electronic communication. A statistical classifier is trained on a set of features extracted from the email content and structure over the training data. After the training, this classifier is used to detect phishing emails from the email stream. These detection schemes differ both in the number and type of features used in the training process. These statistical filters can either be installed on the server or client side. Important maintenance aspects of a machine learning phishing detection scheme is that these filters need to be updated on a regular basis.

Phishing Detection Over Web page Content - Analyzing the structure of the URLs and validating the authenticity of the content of the target web pages. Cantina is a content-based approach to detecting phishing websites based on information retrieval and text mining algorithms. A research team from Google has presented a machine learning technique to accomplish a large scale automatic classification of phishing web pages by analyzing both the URL and the content of the page and achieves 90% accuracy in classifying web pages.

Phishing Detection Using URL analysis – Analyzing only the structure of the links and not the content of the target web pages. They use these features to model a logistic regression filter and show that it has high accuracy in detecting phishing emails.

3. PHISING DETECTION ALGORITHM: PHISNET-NLP

It makes use of all the information present in an email, except attachments, to ascertain(make sure of) which class it belongs to: phishing or legitimate. The first step in the protocol is parsing: PhishNet-NLP accepts an incoming email from the MTA and proceeds to parse it into its constituent components: header, links and text. If the email is HTML encoded, as indicated by the header, we further decode the HTML email body to plain text to perform further analysis. Having obtained the header, links and text, we proceed to analyze each component through their respective classifiers as discussed below. PhishNet-NLP then proceeds to perform majority voting on the scores obtained from the header, link and text analysis classifiers to determine whether an email is legitimate or phish.

The reason for using majority voting as opposed to considering certain weight factors for each of the individual classifiers is to assign an equal importance to each of the classifiers. The majority voting approach has better coverage (accuracy) than that of each individual classifier whenever each classifier in the combination has better than a 50% coverage (accuracy).

Input: *SMTP server name, user name, password*
Output: *Label for each email through classifier: Phishing or Legitimate*

```
1 Fetch email from SMTP server
2 if (new email downloaded) then
3       foreach email e do
4               header hd = extractHeader();
5               if (hd indicates that e is HTML encoded) then
6                       decodedEmail dE=HTMLDecode(e);
7               end
8               parsedEmail pE = emailParser(dE);
9               headerScore = headerAnalysis(header);
10              linkScore = linkAnalysis(links);
11              textScore = textAnalysis(text);
12              cs = combineScore(headerScore, linkScore, textScore);
13              if cs ≥ 2 then
14                      Output Label: Phishing
15              end
16              else
17                      Output Label: Legitimate
18              end
19      end
20 end
```

PhishNet-NLP: Phishing Detection Algorithm

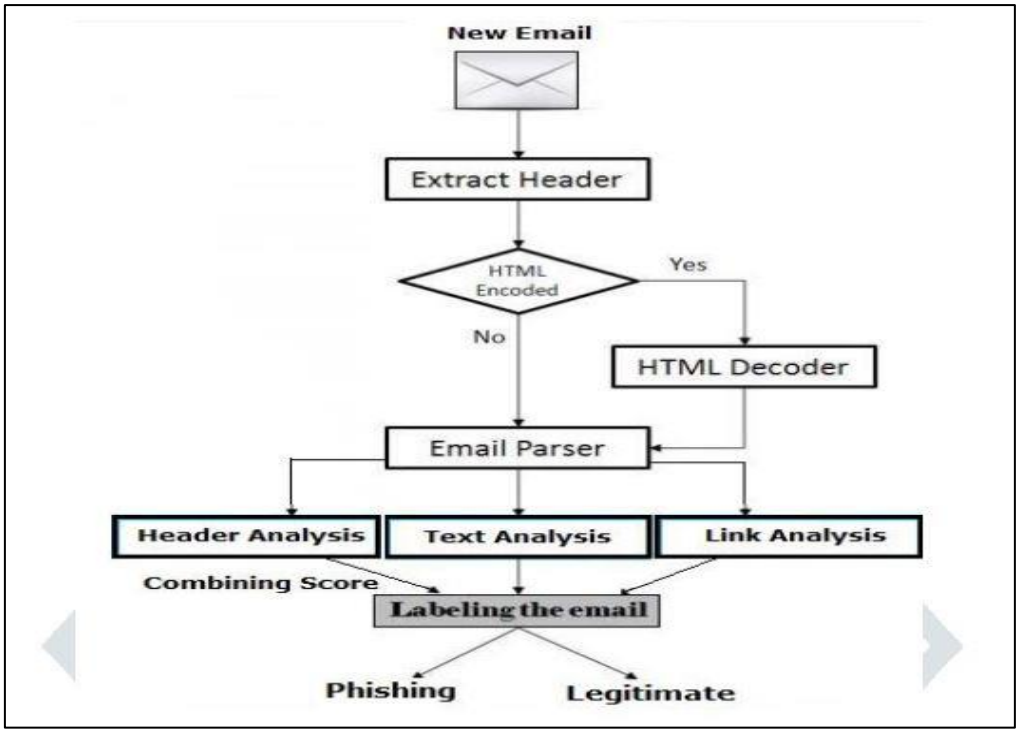

Fig: PhishNet-NLP Phishing Detection Algorithm

4. TEXT ANALYSIS

The goal of email text analysis is to classify the email into two classes: informational and actionable. This is done by analyzing the email text and giving a score to the email called Textscore. The overall approach of PhishNet-NLP is designed for maximum flexibility and efficiency.When the "context" information of an email is available, PhishNet-NLP will use the context to generate a score called Contextscore for the email as well. When the context option is used, then the two scores - the Contextscore and the Textscore are combined logically. To generate the Textscore of the email, we employ a semantics-based method which are as listed.

Stopwords- The aim of stopword removal is to remove common words (such as "the", "a", "an", "in"). For this purpose a stopword list is used. In natural language processing, useless words (data), are referred to as stop words.Applicable when both indexing entries for searching and when retrieving them as the result of a search query.

Stemming- The goal of stemming is to reduce each word form to its root or stem. For example, the verb acting is reduced to act. A popular program for stemming is the Porter Stemmer.

Normalization

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.
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:

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

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.

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.

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.

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

Context Score **-** Context analysis in NLP involves breaking down sentences into n-grams and noun phrases to extract the themes and facets within a collection of unstructured text documents. Through this context, data analysts and others can make better-informed decisions and recommendations, whatever their goals.

1) TF-IDF - Scheme converts a vector of words to a vector of real values using the product of term frequency and inverse document. In information retrieval, TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection.

2) TOKENIZATION - Task of chopping a character into pieces, called as token, and throwing away the certain characters at the same time, like punctuation. It is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. In the

process some characters like punctuation marks are discarded. The tokens become the input for another process like parsing and text mining.

5. HEADER ANALYSIS

In this classifier, we perform analysis on the data from the extracted headers to determine whether the email is phish. First, the user is asked to input his/her other email addresses that forward emails to this current email address and this information is stored.

1) Extracting the data:
We extract the FROM and DELIVERED-TO fields from the header. Then, we extract the RECEIVED FROM field(s) as follows. We look at the received from fields in order, starting with the first such field and then the next such field if present and so on.
If the Received From section of the email contains a DKIM signature, we store the Signing Domain Identifier [SDID]. Otherwise, if there is a Received-SPF field below a Received From field, then first we store the Received From field. Additionally, if the SPF query returns "pass," and if the domain in the From Field accepts an IP address as a permitted sender in the Received-SPF field, we perform an NSLOOKUP on this IP address, and store the domain name corresponding to this IP address in the variable SPFQuery. Otherwise, we store the RECEIVED FROM field.

2) Verifying the data:
If the first Received From field has the same domain name as the FROM FIELD or LOCALHOST or ANY FORWARDING EMAIL ACCOUNT, or if the NSLOOKUP on the IP address of the permitted sender in the Received-SPF field yields the same domain name stored in the variable SPFQuery,
then this email is legitimate. Otherwise, if the first Received From field has the same domain name as the user's current email account's domain name, then we look at the next received from field. The justification for this is provided in the security analysis of our scheme.Otherwise, we mark the email as phishing.

6. LINK ANALYSIS

In this classifier, our objective is to determine whether the URLs present in the email point to the legitimate website that the text in the body of the email claims. We extract all domains from the links in the email in an array (let this array be called DOMAINS). The linkAnalysis() classifier assigns an email a score of 1 for phishing and 0 for legitimate as follows

- If the length of DOMAINS is 0 (no links), the email is legitimate.
- If the email has more than 10 distinct words, we calculate the top four terms in the email using the TF-IDF scores. The IDF value of a word can be obtained by either doing a Google search for the word and obtaining the number of web pages in which it appears. But Google discourages frequent automated searching we used the email database itself to estimate the IDF value.
- Otherwise, if the total number of distinct words in the email is less than 10, then we Google search each domain. If all domains appear in the top 30 results returned by the Google search, then we mark the email as legitimate, otherwise phishing. The reason for insisting on 10 words as a threshold is the very small likelihood of obtaining at least four content words in a text fragment that is shorter.

Combining Scores of the Three Classifiers: Recall that a score of 1 represents phishing and 0 stands for legitimate. If the combined score of the three classifiers (header, link and text) is ≥ 2, PhishNet-NLP labels the email phishing, otherwise it labels it legitimate.

7. ANALYSIS OF PHISHNET-NLP ALGORITHM

PhishNet-NLP is very efficient technique in which it takes each individual part of email and perform its separate analysis and then combine it to get better results. In this case, user should be able to distinguish between phishing and legitimate email. by identifying features such as whether email is asking for urgent reply, or contains some monetary information, asking for money, asking for your personal details, or texts like wining prices etc. Using PhishNet-NLP we are able to increase coverage by about 18% for the phishing emails while obtaining higher accuracy.

Header analysis classifier deals with email forwarding issues and also account for the differences in the headers based on whether the email is sent from a mobile device or relayed by multiple servers in the user's domain. Header analysis examines DKIM (DomainKeys Identified Mail) signatures and SPF (Sender Policy Framework) fields when available.

Our results show that all three classifiers satisfy the minimum threshold needed for helping to mprove the combined classifier since they are all above 50% in coverage and accuracy. The relatively lower percentage of phishing emails detected by textAnalysis() in the two big mail boxes is explained by the imprecision of NLP tools and the three types of emails: foreign language, emails with unusable text, and emails with tables and pictures and insufficient text that we encountered. Also, in each individual mailbox, the 2nd run produced an increased phishing detection by the textAnalysis() classifier and a small increase in the overall phishing detection. This is a direct consequence of the effect of the Context Score, which was not available in the first runs, but available in the 2nd runs after the first runs assigned scores to each email in the database. We could have obtained a higher detection rate on the first run of textAnalysis() by using the previous context of the first N emails when processing email $N + 1$. However, we preferred to keep a fixed context for analysis of each email rather than a growing context, since in this case our results are insensitive to the order in which emails are processed.

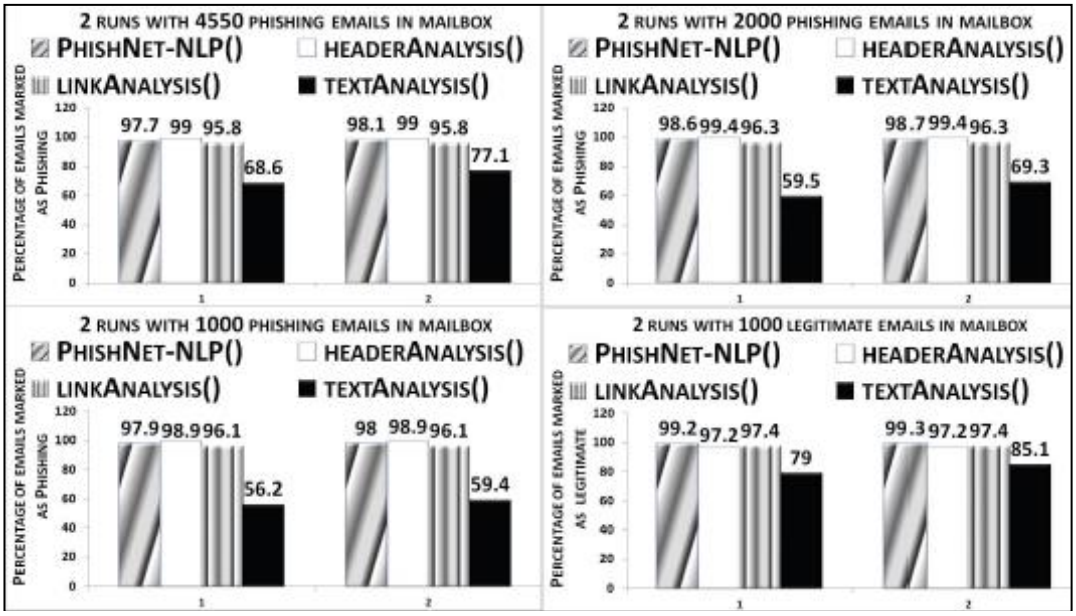

Fig: Results

8. SECURITY ANALYSIS

Analyze the security of this scheme against several scenarios and discussion of various aspects.

Is textAnalysis() or linkAnalysis() Redundant? Observe that while the headerAnalysis() classifier alone shows very high coverage and high accuracy, the importance of link and text analysis stems from the fact that a sophisticated phisher can manipulate the originating "Received From", "From" and the "Delivered To" information completely. Link and Text analysis are very important and provide robustness to this scheme.

Attacks Based on Knowledge of Our Scheme The reader might think that a phisher can analyze how our detection algorithm works and then design a phishing email to fool PhishNet-NLP. But results from the LinkAnalysis show that it is very difficult to create a fraudulent link to bypass LinkAnalysis. Moreover, unless the phishers have hacked into the mail server or the user's account, they would not have access to the context of the user's mailbox. Hence, it is likely that Context Analysis will also play a part in detecting such an email.

Insider Attacker When someone hacks into an account in some domain and uses a friend list to attack any user in the same domain, headerAnalysis() will fail to detect this. But even in such a case, PhishNet-NLP can use the linkAnalysis() and textAnalysis() to mark the email as phishing since the intent of the email is still to steal sensitive information by asking the user to click on a link for a malicious website. This even works for the scenario when user A's account is hacked and user A receives a phishing email, for example, if A's sensitive information is stored in an encrypted form.

Foreign Language Email or Emails with Insufficient Text As of the present design, emails in foreign languages or emails with insufficient text (only links or attachments) present a challenge to the textAnalysis() classifier which leads to a low phishing detection rate by the textAnalysis() classifier. However, we were able to offset this to a certain extent by using context analysis to correctly identify the email as phishing.

9. CONCLUSION

In this paper, we presented a phishing detection scheme called PhishNet-NLP. This is the first scheme to utilize natural language based techniques and context information when available to detect phishing. PhishNet-NLP operates by inferring the "intention" of the email - whether it is informational or actionable. Our phishing detection rate is at least 97% with very low false positives. Another novel feature in PhishNet-NLP is that we utilize all of the information available in an email, namely, the header, links and text of an email. This scheme operates in the default mode and does phishing detection in the absence of any history. The novelty lies in the fact that when prior history is available, our scheme takes advantage and improves the detection capability. Finally, this scheme is designed to detect phishing at the email level rather than to detect fraudulent, masqueraded websites thereby protecting the user from the start.

REFERENCES

[1]. Detecting Phishing Emails the Natural Language Way https://link.springer.com/chapter/10.1007/978-3-642-33167-1_47

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

[3]. Zhang, Y., Hong, J., Cranor, L.: Cantina: a content-based approach to detecting phishing web sites.

[4]. www.google.com

[5]. https://www.researchgate.net/publication/278689575_Detecting_Phishing_Emails_the_Natural_Language_Way

[6]. Rakesh Verma1, Narasimha Shashidhar2, and Nabil Hossain3: Detecting Phishing Emails the Natural Language Way