# IDENTIFICATION AND DETECTION OF PHISHING EMAIL USING NATURAL LANGUAGE PROCESSING TCHNIQUES

**Miss. Amruta V. Mehendale**

DES's Navinchandra Mehta Institute Of Technology and Development, Dadar(W).

Mail Id: amrutamehendale95@gmail.com

***Abstract:*** **Email is still the most commonly used medium to launch phishing attacks [1]. The number of phishing emails is increasing tremendously day by day. Phishing email resulting in financial losses for organizations and annoying individual users. It poses a serious threat to the Internet economy. This need to be detected and prevented from attackers for not to elicit victim's sensitive information. The paper presents a comprehensive natural language-based scheme to detect phishing emails using features that are invariant and fundamentally characterize phishing. This scheme utilizes all the information present in an email, namely, the header, the links and the text in the body. Text in the email body contains some common features such as, a mention of monetary incentive and a sentence inducing the recipient to reply. Such texts are analysed using algorithm and distinguish between "legitimate" and "phishing" emails. To this end, it incorporate natural language techniques in phishing detection. This is crucial to prevent the victim from clicking any harmful links in the email. The implementation called PhishNet-NLP[2], 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.**

## *I. INTRODUCTION:*

Email is most widely used technique to communicate and transfer data from one user to other and that is why it is most popular medium to launch phishing attacks. Attacks are typically carried out via communication channels such as email or instant messaging by attackers masquerading as legitimate and trustworthy entities. According to Kaspersky Lab's "Spam and Phishing in 2017" report [3], criminals have been following a global agenda by using hot topics such as FIFA 2018 and Bitcoin to fool users and steal their money or personal information in the last 12 months. Every time attackers use different techniques to fool customers by analysing their search result and send relevant emails to attract them. These attackers pose as some trustworthy entity and try to make the victim voluntarily give his/her sensitive details which may be personal or belonging to an organization. These sensitive details can be usernames, passwords, bank account details or can even be information used to create fake Identities of the victim. To identify this email phishing, the primary contribution in this paper is a comprehensive and effective natural language-based phishing detection scheme. 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 Language Processing (NLP) and WordNet. This scheme uses the information present in the email header, text in the email body and the links embedded in the email. 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. PhishNet-NLP operates between a user's MTA and MUA and processes each arriving email for phishing attacks which prevents the user from clicking any harmful link in the email. This approach contrasts with schemes that analyse the target websites for authenticity.

## *II. DEFINITIONS TERMS AND TOOLS:*

## 2.1 Natural Language Processing

NLP is an area concerned with enabling the computer to derive meaning from the language spoken by humans. In this, machine interprets the important elements of the human language sentence, such as those that might correspond to specific features in a data set, and returns an answer. The idea is to make a computer similar to human in a way that it can process data and instructions in a meaningful manner. Due to the inherent ambiguity and rich structure of natural languages, this approach to email text processing employs the following NLP techniques: lexical analysis, part-of-speech tagging, normalization of words to lower case, , named entity recognition , stemming and stopword removal.

2.1.1. Lexical analysis - Lexical analysis [4] splits the program source code into substrings called tokens and classify each token to their role (token class). The goal of lexical analysis is to split the email into sentences and each sentence into words.

2.1.2. part-of-speech tagging [5]- It tags each word in the email and identify whether that word is noun, verb, article, adjective etc.

2.1.3. normalization of words to lower case-In this, words are converted to lower case in a normalization phase.

2.1.4. named entity recognition- It tags the named entities in the email, which are nouns that name either person, location or organization or any other entity.

2.1.5. 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 [6].

2.1.6. stopword removal -The aim of stopword removal is to remove common words such as it, a, an, the,etc. For this purpose a stopword list is used.

## 2.2. WordNet

WordNet is one of the largest and most widely used electronic dictionaries, thesauri. It has been used for many natural language processing tasks, including information retrieval, word sense disambiguation and question answering. According to Fellbaum [7], WordNet is a database that combines the functionalities of a dictionary and a thesaurus. The words in WordNet are arranged hierarchically. It contains the synonyms as well as the meanings of a word. Here is one example of WordNet to understand it better.

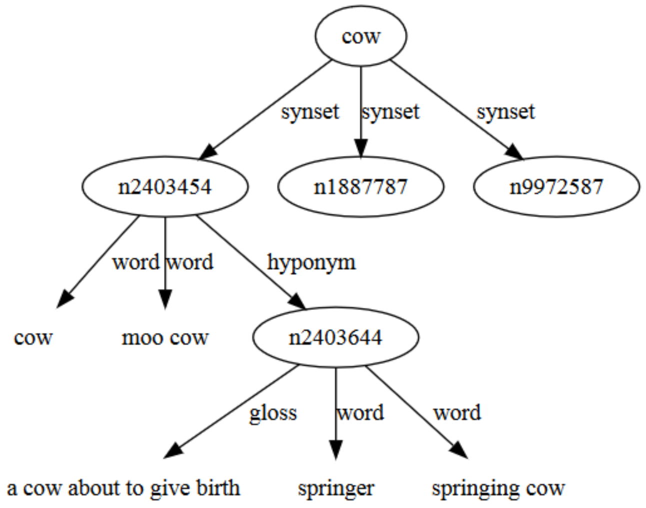

Fig 1. WordNet Hierarchy
(Source: WordNet HTTP API)

## 2.3. TF-IDF

Term Frequency is used to calculate how many number of times the term appear in particular document or set of texts. Inverse term frequency (IDF) is to compute the frequency of the same word throughout the whole dataset. Together, these two quantities can be used as a measure of how important a term is to a particular document. Calculate term frequency(TF) in each document. Iterate each document and count how often each word appears. Calculate the inverse document frequency (IDF): Take the total number of documents divided by the number of documents containing the word. Calculate TF-IDF: multiply TF and IDF together.

### 2.4. DKIM: Domain Keys Identified Mail

In this technique, A sender creates DKIM by signing the email with a digital signature. This signature is located in the message's header. By using an algorithm applied to the content of the signed fields, the sending mail transfer agent (MTA) generates the signature. This algorithm creates a unique string of characters, or one can say a hash value. After receiving the email, the recipient MTA can verify the DKIM[8] signature. The recipient MTA then uses that key to decrypt the hash value in the email's header and simultaneously recalculate the hash value for the email message it received. If these two keys match, then the email has not been altered, giving users some security knowing that the email did originate from the listed domain, and that nothing has modified it since it was sent.

### 2.5. SPF: Sender Policy Framework

SPF[9] is an open standard so that the owner of a domain can provide a public list of approved senders. SPF mention all the received from fields. So an email can pass SPF regardless of whether the from address is fake.

### 2.6. NSLOOKUP

NSLOOKUP is the name of a program that lets an Internet server administrator or any computer user enter a host and find out the corresponding IP address. It will also do reverse name lookup and find the host name for an IP address you specify.

## *III. PHISHING DETECTION ALGORITHM: PHISHNET-NLP*

PhishNet-NLP is algorithm make use of all the information present in the email and identify which class it belongs to either phishing or legitimate. This algorithm takes all the information, texts, subject, from, to etc except attachments and then perform its task. As said earlier, this protocol applies between a user's mail transfer agent (MTA) and mail user agent (MUA) and processes each arriving email. First it divides the email into three components: header, links and text. If the incoming email is HTML encoded, further decode the HTML email body into plain text to perform further techniques. Once header, links, and text are extracted, now proceed to analyze each component through their respective classifiers. Then this algorithm perform majority voting and for that it takes scores obtained from each component(header, links, text) to determine whether particular email is phishing or legitimate.

### 3.1. PhishNet NLP Algorithm Flow Diagram

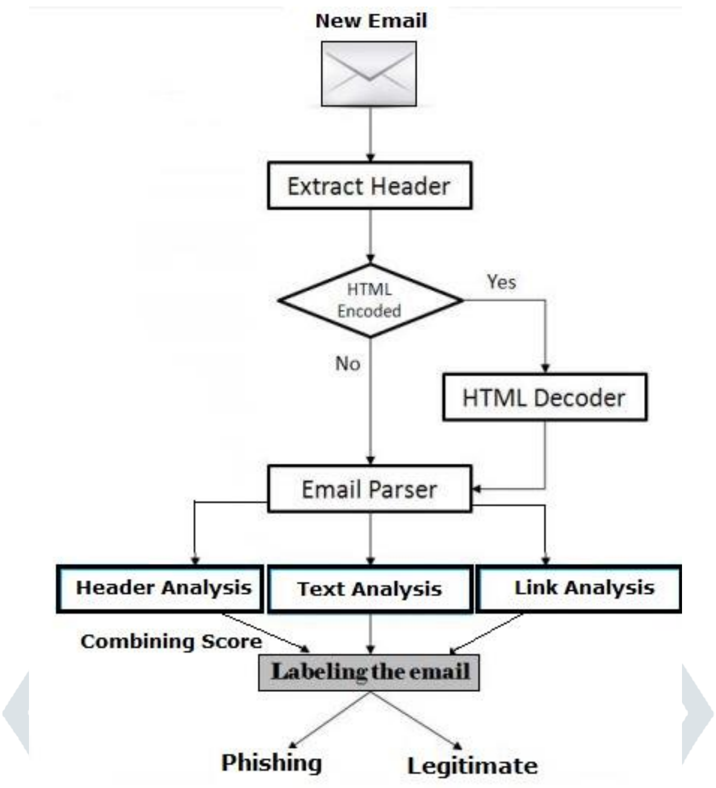

Fig 2: Phish-Net NLP Detection Algorithm Flow

### 3.2. PhishNet NLP Algorithm

```
Input: SMTP server name, user name, password
Output: Label for each email through classifier: Phishing or Legitimate
Fetch email from SMTP server
if (new email downloaded) then
    foreach email el do
        header hd = extractHeader();
        if (hd indicates that el is HTML encoded) then
            decodedEmail dE=HTMLDecode(el);
        end
        parsedEmail pE = emailParser(dE);
        headerScore = headerAnalysis(header);
        linkScore = linkAnalysis(links);
        textScore = textAnalysis(text);
        cs = combineScore(headerScore, linkScore, textScore);
        if cs ≥ 2 then
            Output Label: Phishing
        end
        else
            Output Label: Legitimate
        end
    end
end
```

### 3.3. Explanation and Implementation of Algorithm:

Now, Let's understand NLP techniques used in each component and how it help us to identify phishing email.

### 3.3.1. Text Analysis:

Text analysis[10] is a major component as it contains actual text message of email. To go to the final stage of classifier, one must apply pre-processing technique on email text messages so that it can filter out necessary words in a proper format to identify classifier. Each NLP problem requires a tailored approach to determine which terms are relevant and meaningful.

#### 3.3.1.1. Normalization:

Sometimes terms are randomly capitalized, others are in all caps so one need to perform normalization. Perform all this pre-processing techniques using python. Next remove all punctuations since "Tomorrow" and "Tomorrow?" refer to the same word. In addition, remove all whitespaces, line breaks, tabs into single space. Convert all words to lower case to make it more simpler.

Any spam email is identified by analysing its features. The main aim is to exploit the common features[11] among such phishing emails such as no mention of the victim's name in the email, a mention of some amount of money, provide some links to click and a sentence asking for a reply, sense of urgency etc. Every feature has its own set of words.

1) No mention of victim's name:

Sometimes attacker does not make use of victim's name. Instead it make use of phrases like "Dear Friend", "Hi Dear", "My Dear Beneficiary" and various such phrases which lack the recipient's name. To find whether attacker have mentioned user's name or not there is one method. Get the user name from their account to analyse, and make three sets for First Name, Middle Name(if any) and Last Name as F,M, and L respectively. Now N is set which is calculated by finding Union and cross product of these three sets. First it take one set at time, then take two of them at a time and then take three of them at a time in all possible permutations.

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

Fig 3: Possible Permutations on feature No mention of victim's name
(Source: Detection of email phishing)

Now there are all the possible name variants of the user, so find whether her name is present in the mail or not.

2) Mention of some amount of money:

If email contains any amount then it can be identified by some number preceded with currency symbol such as $ or currency name like Rs, dollar. In the email analysis attackers have used mostly popular international currencies. Make set of all these currency symbol and names.

3) Sentence asking for a reply:

A reply inducing sentence contains a word or phrase which asks the user to reply to the email. The words for such messages are stored in a set R {write, contact, get_back, reply, response, forward, send, hear, click, follow, visit, go, update, apply, submit, confirm, cancel, dispute, enrol} Let set R be a set of all such words which ask for a reply from the users.

For set X contain 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 any natural number $i \geq 1$, let $\text{hypo}^{i}(\text{Synset}(R))$ denote the union of all the synsets reached by following upto i hyponymy links from the synsets in Synset(R).

4) Sense of urgency:

This feature asks for replies as soon as possible or urgently. Let U = {now, nowadays, present, today, instantly, straightaway, straight, directly, once, forthwith, urgently, desperately, immediately, within, inside, soon, shortly, presently, before, ahead, front} These are the words that mention sense of urgency.

### 3.3.1.2. Removing stop words:

In English language some words does not contribute much to the meaning of a phrase. Some words which are less important for analysing need to be removed to focus only on main words. Even Google search engine use this technique to remove unnecessary stop words and collect only the keywords from it. For this, a popular python library NLTK( Natural Language Toolkit) can be use. This library provides some common stop words. By applying this library, compare stop words from this toolkit to our email message. This saves time of pre-processing step.

### 3.3.1.3. Stemming:

If an email message contains same words with various suffix such as "distribute", "distributing", "distribution", "distributor" etc, replace these words with one common word "distribut" by this pre-processing technique called stemming. Stemming technique is also available in NLTK toolkit called Porter Stemmer.

All these pre-processing techniques can be combined to make one single code by taking it into one function. After executing this function one will get output which contains all lower case letters, features taken into variables and only keywords are taken necessary to identify whether email is phishing or legitimate. At this output pre-processing is done. For every keyword, appropriate verb is attached at the end of the word in WordNet. Now, calculate text score[12] on pre-processed output for all these features ,get the final text score which can later combined with context score. Calculating Text Score use formula,

For any word r ε SR, let

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

Fig 4: Final Text Score Formula
(Source: Detection of email phishing)

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.

### 3.3.2 Context Score:

PhishNet-NLP will use the context to generate a score called Context score for the email. It is the analysis score of email comparing other saved emails of the user which includes sent and received emails. Extracted important words called keywords used for tokenization which takes n-gram words. Then TF-IDF statistics is performed on those n-gram words which finds the frequency of words which appeared frequently but in very limited emails.

### 3.3.2.1. Tokenization:

Tokenize individual terms and generate bag of words model. Here tokenize every sequence of n terms called n-grams. Tokenizing adjacent pairs of words called bigrams. To get best of both, let's tokenize unigrams and bigrams. For example "you will win price" .This sentence will be "You", "will", "win", "price", "you will", "will win", "win price".

### 3.3.2.2. Implementing the TF-IDF statistics:

After applying tokenization strategy, next step is to compute n-grams frequency using some statistics called TF-IDF. In this paper, it is already mentioned what is TF-IDF and how it calculates. Now let's see how it is applied on finding phishing email. One statistic called term frequency (tf) tallies the occurrences of each n-gram. However, some n-grams will definitely show up often in any given Email messages, while others rarely appear in the overall document but show up frequently in certain *subsets* of messages such as spam. Here frequency of the term is used in the collection for weighting and ranking. Rare terms are more informative than frequent terms. Here low positive weights are needed for frequent terms and high weights for rare terms. Therefore, to emphasize more interesting set of n-grams, down weighs the term frequency with inverse document frequency (idf).

Combining these two statistics yields the tf-idf statistic:

$tf\text{-}idf(t,i)=tf(t,i)\times idf(t)$

$=tf(t,i)\times log(M/m_t)$

where $tf(t,i)$ is the term frequency for term $t$ in the $i^{th}$ training example, $M$ is the total number of training examples, and $m_t$ is the number of training examples that contain the term $t$.

### 3.3.3 Header Analysis:

Generally one can understand email header contains senders, receiver's address and subject, but apart from that there are many more information that email header contains.

Email Header consist of sender, receiver, subject, Message ID, IP address, domain name, mail user agent, Servers in transit. All these information is not seen directly by user.(If user want to see header information then there is provision to get the complete header). This information helps to identify whether email is coming from legitimate user or it is a fake entity. In this, make use of DKIM and SPF authentication techniques as seen earlier.

#### 3.3.3.1 Extracting Data:

From header information extract FROM address, DELIVERED-TO and RECEIVED-FROM fields from header. RECEIVED-FROM fields can be one or many because email is not directly transfer from sender to receiver. It traverse through various intermediate destination and those destination keep on adding its own server name, IP address also the name of the mail server itself and pass it to next. So it starts with first field and the next such field if present and so on. If the very first RECEIVED-FROM fields contains DKIM signature then store Signing Domain Identifier(SSID) which is a single domain name that is mandatory payload output of DKIM, refers to the identity claiming some responsibility for the message by signing it. Otherwise, if there is RECEIVED-FROM field next to it then store that field. Fetch its IP Address and get its domain name through NSLOOKUP.

#### 3.3.3.2. Verifying Data:

If the first Received From field has the same domain name as the FROM field or the IP address of that is same mentioned in NSLOOKUP. If those domain names are valid then this email is legitimate. Check this for all received from field one by one and check for its authenticity. If anyone becomes false then email is marked as phishing email.

### 3.3.4. Link Analysis

Generally link in the email is related to the email text. For example if email text contains information like bank, deposit, amount then definitely that link is about bank sites or any other investments scheme or any monetary site etc. For link analysis make use of dataset which consist of 203,979 potential phishing websites collected from a large spam-based URL provider, a large anti-phishing company, and a number of other feeds including private companies, security companies, and financial institutions. Then if email contains any link then that link is checked with this dataset which contains blacklisted and whitelisted links. If any match found then value will be 1 that means its fake, else t will be 0 means its legitimate.

### 3.3.5. Combining Scores of the Three Classifiers:

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

## *IV. ANALYSIS OF PHISHNET-NLP ALGORITHM:*

1) 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. This algorithm is applied on email header, email text, email link classifiers but not on attachments which also a part of email. If you click on any attachment to download, virus, malware can also get downloaded on your PC and it can easily destroy your computer's data. So downloading of such attachments can also be avoid through this PhishNet-NLP technique. Before applying this technique directly, first identify

whether email contains attachments or not. Generally emails have attachments symbol through which one can identify. If email have attachment, then see the attached file's extension. Extensions should not be .exe, .js or .zip. Do not download files if file has these extensions. Mostly attached files have word, excel, power point, PDF, notepad files etc which can be harm or cannot b. That can identify through PhishNet-NLP technique. First open that file in the background, by taking its content to the notepad file and then apply Text Analysis on its content, take information about asking for any security details, analyse for malware and viruses. If algorithm found such kind of information then user should not download such attachments. If file does not contain such contents, then users can download the file.

2) As this phishNet-NLP technique is more accurate and efficient, but there can be 1% chances of failure. In that case, user should know few things which email is phishing and which email is legitimate. To understand this, user should mainly focus on email 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. These features helps user for phishing email. Also it can check header whether this email is coming from legitimate user or not. This can be done by user. User just have to click on email and click on symbol present near reply. Then click on show originals. Then you will get complete information of email header. So user also have to notice few things.

## *V. CONCLUSION:*

In this paper, the scheme detected email phishing detection technique through Phish-Net which make use of natural Language processing Technique to detect. Several techniques were used for text analysis, header analysis and link analysis. By combining all three classifiers one get the best results to identify whether email is phishing or legitimate. Here conclusion is that this technique gives more accuracy than any other techniques as this technique use NLP which itself contain many other techniques as WordNet, TF-IDF, stopword removal, POS tagging stemming etc and analyse each classifier where, they get combined result and based on threshold value it identify phishing email.

This scheme is very efficient because, it directly give result whether email is phishing or legitimate directly in the inbox.

## *VI. 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] Rakesh Verma, Narasimha Shashidhar, Nabil Hossain, Detecting Phishing Emails the Natural Language Way. https://link.springer.com/chapter/10.1007/978-3-642-33167-1_47

[3] Kaspersky Lab report: using hot topics such as FIFA 2018 and Bitcoin to fool users and steal their money or personal information Available: https://usa.kaspersky.com/about/press-releases/2018_fifa-2018-and-bitcoin-among-2017-most-luring-topics

[4] Lexical analysis- https://medium.com/dailyjs/gentle-introduction-into-compilers-part-1-lexical-analysis-and-scanner-733246be6738

[5] part-of-speech tagging -https://medium.freecodecamp.org/an-introduction-to-part-of-speech-tagging-and-the-hidden-markov-model-953d45338f24

[6] Porter, M.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)

[7] Fellbaum, C. (ed.): WordNet An Electronic Lexical Database. MIT Press (1998)http://www.lexicadb.com/lxserver/wn_http.html

[8] Hansen, T., Crocker, D., Hallam-Baker, P.: Domainkeys identified mail (dkim) service overview (2009), https://tools.ietf.org/html/rfc5585

[9] An overview of the Sender Policy Framework (SPF) as an anti-phishing mechanism S. Görling - Internet Research, 2007 - emeraldinsight.com http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.670.3345&rep=rep1&type=pdf

[10] Using natural language processing to build a spam filter for text messages http://inmachineswetrust.com/posts/sms-spam-filter/

[11] Identification and detection of phishing emails using natural language processing Techniques, S Aggarwal, V Kumar, SD Sudarsan - Proceedings of the7th …2014