2018 12th IEEE International Conference on Semantic Computing

Detecting Phishing Attacks Using Natural Language Processing and Machine Learning

Tianrui Peng
Columbia University
New York, NY, USA
tp2522@columbia.edu

Ian G. Harris
University of California Irvine
Irvine, CA, USA
harris@ics.uci.edu

Yuki Sawa
University of California Irvine
Irvine, CA, USA
yukisawa@gmail.com

***Abstract*—Phishing attacks are one of the most common and least defended security threats today. We present an approach which uses natural language processing techniques to analyze text and detect inappropriate statements which are indicative of phishing attacks. Our approach is novel compared to previous work because it focuses on the natural language text contained in the attack, performing semantic analysis of the text to detect malicious intent. To demonstrate the effectiveness of our approach, we have evaluated it using a large benchmark set of phishing emails.**

I. Introduction

In contemporary society, the security of private information is a major concern of every person. Social engineering attacks are dangerous threats that aim at using human interaction manipulate people into exposing their confidential information or performing inappropriate actions.

Phishing is a type of social engineering attack that focuses on gaining sensitive information by disguising as a trustworthy entity. Electronic communications, such as email or text message are common platforms for delivering phishing attacks. Phishing has been shown to be an effective attack over the years, deceiving a broad range of people [4]. Attackers are usually disguised as popular social websites, banks, administrators from IT departments or popular shopping websites. These emails may lure users to click on links to initiate malware downloads, or enter personal information into a malicious website which has a similar look to a legitimate one. Most automatic phishing email detection approaches rely on email metadata, data associated with emails which is not related to the semantic meaning of the text message. Several approaches examine the URLs contained inside the message [2]. There are several phishing detection approaches which evaluate text by searching for the presence of specific words in each sentence [9], [10], [1]. Previous work [8] has also employed syntactic parsing to infer malicious intent.

Our approach performs a semantic analysis of the text transmitted by the attacker to verify the appropriateness of each sentence. A sentence is considered to be malicious if it inquires sensitive information or commands a performance of action that might expose personal information. Natural language processing (NLP) techniques are applied to parse each sentence and identify the semantic roles of important words in the sentence in relation to the predicate. Based on the roles of each word in the sentence, our approach determines if the sentence is a question or a command. The potential topics of questions and commands are extracted by finding (verb-direct object) pairs. Then each pair is evaluated by whether it is contained in a *topic blacklist* of malicious pairs. We use supervised machine learning to generate the blacklist of malicious (verb-direct object) pairs based on the pairs found in a training set of phishing and non-phishing emails.

II. Detection Algorithm

Our system, which we have named *SEAHound*, processes a document, one sentence at a time, and returns True if the document contains a social engineering attack. The algorithm for detecting phishing emails is shown in Figure 1.

```
1.  define SEAHound(text)
2.      bad, urgent, generic = False
3.      foreach sentence s in text
4.          bad |= BadQuestion(s) OR BadCommand(s)
5.          urgent |= UrgentTone(s)
6.          generic |= GenericGreeting(s)
7.          link = LinkAnalysis(s)
8.          if link
9.              return True
10.     if majority(bad, urgent, generic)
11.         return True
12.     return False
```

Fig. 1. SEAHound Algorithm

The email algorithm in Figure 1 evaluates each sentence (lines 3-9) to determine if it exhibits four characteristics: 1) malicious question/command (line 4), 2) urgent tone (line 5), 3) generic greeting (line 6), and 4) malicious URL link (line 7). An email is considered to be malicious if a malicious link is found (lines 8-9) or if at least 2 of the remaining three characteristics are found in the email (lines 10-11).

The $LinkAnalysis$ step which verifies the validity of a URL is performed using the Netcraft Anti-Phishing Toolbar which is a commercial tool and has been shown to be effective in previous studies [3]. Although the link analysis provided by Netcraft is effective, it is limited to URL analysis, so it cannot

0-7695-6360-0/18/$31.00 ©2018 IEEE
DOI 10.1109/ICSC.2018.00056

IEEE computer society

detect social engineering attacks which do not include URL links. Our results demonstrate that using Netcraft for URL analysis alone is inferior to our approach which also performs semantic analysis of the text.

III. Machine Learning for Blacklist Generation

The identification of malicious questions and commands depends on the the existence of a **topic blacklist** which is a list of (verb-direct object) pairs whose presence in a question or command suggests malicious intent. To generate the topic blacklist we use machine learning, developing a Naive Bayes classifier which is designed for multinomially distributed data, and is commonly used for text classification. We used the *MultinomialNB()* function from the Scikit-learn Python library [7] which implements this algorithm. This algorithm produces a prediction label for each (verb-direct object) pair, and generates a confidence score for the prediction. The range of the confidence scores is 0 to 1, with a confidence score of 1 indicating certainty.

We used 1000 phishing emails from the Nazario phishing email set [6] and 1000 non-phishing emails from the Enron Corpus [5] as our training set. We tested our results on all 5014 emails in the Nazario phishing email set and on 5000 non-phishing emails in the Enron Corpus. Before applying machine learning, we used the Stanford typed dependency parser to extract all (verb-direct object) pairs from all sentences by identifying the "nsubjpass" and "dobj" dependencies in the dependencies found in each sentence.

After training, we considered a (verb-direct object) pair to be malicious if its certainty exceeded a threshold. We experimented with different confidence cutoffs to tradeoff the need for high accuracy and for low false positive rate. In the end, we determined that a pair is malicious if its confidence score was 0.9 or higher. Based on this threshold we identified 636 (verb-direct object) pairs which were used as the topic blacklist.

IV. Experimental Results

In order to properly evaluate our approach for false positives and false negatives, we have utilized two email datasets. We have used a publicly available phishing email set compiled by Jose Nazario [6]. For the legitimate email corpus we have used the Enron Corpus [5]. Some of the phishing emails from the Nazario set contained only images with no text outside of the images. We ignored phishing emails containing only images and used all remaining 5009 phishing emails in the Nazario set. Our test set also included 5000 non-phishing emails from the Enron Corpus.

For comparison, we have evaluated the test corpus with our algorithm and with Netcraft alone, which only detects phishing URL links. Our algorithm was implemented as Python scripts and are executed on an Intel Core i7 processor with 8Gb of RAM.

In Table I we report five values for each approach, true positives (TP), false positives (FP), false negatives (FN), precision, and recall. Precision and recall are defined as follows,

$$precision = \frac{TP}{TP + FP}$$

and

$$recall = \frac{TP}{TP + FN}$$

. Table I shows that our approach, when compared to Netcraft, results in a reduced number of false positives at the expense of the number of false negatives. The decrease in false negatives shows that semantic information is a useful indicator to identify phishing attacks. However, the approach as presented provides a precision of only 95%.

V. Conclusions

We present an approach to detect targeted phishing email attacks. Our approach relies on analysis of the text, rather than metadata which might be associated with emails. As a result, our approach is effective for detecting phishing emails which are composed of pure text. Our results on phishing emails demonstrate significantly improved recall which demonstrates that semantic information is a strong indicator of social engineering.

References

[1] Aggarwal, S., Kumar, V., and Sudarsan, S. D. Identification and detection of phishing emails using natural language processing techniques. In *Proceedings of the 7th International Conference on Security of Information and Networks* (2014), SIN '14.

[2] Chen, J., and Guo, C. Online detection and prevention of phishing attacks. In *First International Conference on Communications and Networking in China* (Oct 2006).

[3] Cranor, L., Egelman, S., Hong, J., and Zhang, Y. Phinding phish: An evaluation of anti-phishing toolbars. Tech. Rep. CMU-CyLab-06-018, Carnegie Mellon University CyLab, November 2006.

[4] Jagatic, T. N., Johnson, N. A., Jakobsson, M., and Menczer, F. Social phishing. *Commun. ACM 50*, 10 (2007), 94–100.

[5] Klimt, B., and Yang, Y. *The Enron Corpus: A New Dataset for Email Classification Research.* 2004.

[6] Nazario, J. The online phishing corpus. https://monkey.org/~jose/phishing/, 2005. Accessed: 2016-09-13.

[7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[8] Sawa, Y., Bhakta, R., Harris, I. G., and Hadnagy, C. Detection of social engineering attacks through natural language processing of conversations. In *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)* (Feb 2016).

[9] Stone, A. Natural-language processing for intrusion detection. *Computer 40*, 12 (Dec 2007).

[10] Verma, R., Shashidhar, N., and Hossain, N. Detecting phishing emails the natural language way. In *Computer Security ESORICS 2012*, S. Foresti, M. Yung, and F. Martinelli, Eds., vol. 7459 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012.

	TP	FP	FN	Precision	Recall
SEAHound	4545	239	464	95%	91%
Netcraft	3625	83	1384	98%	78%

TABLE I
Phishing Detection Results