A Method with Pre-trained Word Vectors for Detecting Wordlist-based Malicious Domain Names

Shaoqing Lin1,*, Shangping Zhong, Kaizhi Cheng

College of Mathematics and Computer Science, Fuzhou University, Fuzhou, 350108, China

*m18250150928@163.com

Abstract. In recent years, botnets have used the domain generation algorithm to generate dynamic typified malicious domain names to bypass various detection methods. Given the depth detection model of such domain names, domain names are generally processed by filling and transforming them into a fixed-length one-dimensional vector and then classifying them with poor detection performance. Therefore, this study first divides the domain into a word array and converts it into a word vector using pre-trained word vector models, Embeddings from Language Models. The domain is inputted into the TextCNN model for training classification. From approximately 100,000 data sets, a 94.22% accuracy rate and 6.87% FPR value can be obtained from the training. Compared with previous detection models (i.e., LSTM and CNN), more training and testing are needed, but improvements are made in all indicators.

Keywords: Wordlist-based DGA Domain Name, Depth Detection, Pre-training Model, Word Vector, TextCNN

1. Introduction

Malware poses a serious threat to individuals and businesses[1]. Typical attack methods, such as viruses, phishing emails, and worms, attempt to retrieve private user data, disrupt systems or start unwanted programs. Most of these attacks are likely to be launched over networks [2] and pose a significant threat to any Internet-facing device. To achieve its goal successfully, malware needs to connect to command and control centers (C&Cs). Zombie hosts behind C&C centers and malware on infected machines can run domain generation algorithms that automatically generate hundreds or even thousands of domains.

The malicious domain name generation algorithm (DGA) is divided into many families according to different algorithms. Some of these early algorithmic families seed the infected computer with data parameters, such as time. The domain name generated by this type of character DGA has great randomness making it difficult to coincide with a legal domain name in reality but easy to distinguish from the legal domain name in terms of length, character distribution, and word rationality.

However, the wordlist-based DGA has three main families, namely, Suppobox [4], Gozi [5], and Matsnu [6]. Domain names, such as journeydaughter.net and littledaughter.net, are generated by stitching algorithms based on word dictionaries. Manually distinguishing them may not give correct
conclusions. Thus, typical word malicious domain names can be confused with legitimate domain names.

The focus of this study is to detect the typical DGAs of these three types of characters. In this study, according to the characteristics of typical malicious domain names, the processing of domain names is different from the existing methods. Instead of converting each character into a data input, the domain names are segmented into words and converted into word vector input by the pre-training model. The experiment compares the accuracy rate, F1 score, AUC, and other indicators to obtain the most appropriate typical malicious domain name detection model.

2. Related word

2.1. Malicious DGA

The primary purpose of a DGA is to generate unregistered domain names. Numerous DGA families have two categories, namely, character-based generation and dictionary-based generation [7]. The domain names generated by different families are shown in Tab.1.

| Domain type          | Family      | Domain name                             |
|----------------------|-------------|-----------------------------------------|
| Legal domain name    | From Alexa  | google.com                               |
|                      |             | facebook.com                            |
| Character-based      | tinba       | sjuqlwqrhhx.com                          |
|                      |             | oqckhpisyqll.in                         |
|                      | locky       | qtysmobytagnrv.it                       |
|                      |             | rddipikmrap.us                          |
|                      | dircrypt    | jarvddjzqrmmneqwd.com                   |
|                      |             | swtjyuhuefvl.com                        |
| Wordlist-based       | suppobox    | earnestinechauncey.net                  |
|                      |             | willoughbycalanthe.net                  |
|                      | gozi        | inestimabiler.com                       |
|                      |             | papacriconognitionipro.com             |
|                      | matsnu      | articlestafftellfingermeet.com          |

Tab.1 Sample multiple family malicious domain names

2.2. MoMalicious domain name detection algorithm

A lot of work on malicious domain name detection has been done in recent years. Woodbridge et al. [8] were the first to use the deep learning model to solve such problems. Their experiments showed that their deep learning method performed better than the random forest model with characteristics such as character distribution entropy, and their model achieved great success in identifying most of the DGA families. Since then, others have joined the field and implemented a variety of deep learning models. Several use Woodbridge et al. ’s LSTM model. And provide improvements. Tran et al. [9] considered the class imbalance of DGA data.

Pereira et al.’s WordGraph [10], which aims at the wordlist-based DGA, analyzed the words in the domain name data by using the characteristics of the word dictionary to determine the dictionary used by the DGA. However, this approach can be excessively computational because the domain names need to be split up and the properties of the graph must be calculated, which are impractical.

The TextCNN [11] model proposed by Kim et al. is used in the present study. It has a strong ability to extract shallow features of texts, which is effective and widely used in short text fields, such as search and conversation fields, when the focus is on intention classification.

2.3. Word vector model
The embedding of words and sentences has become an important part of all-natural language processing systems based on deep learning. Words and sentences are encoded in the dense vectors of fixed length to improve the ability of neural networks to process text data. The Word2vec model is derived from a study published by Mikolov et al. [1] in 2013. Its core idea is to obtain the vectorization representation of a word through its context. Another static word vector model is GloVe[13], which uses a co-occurrence matrix and considers local and global information. This study uses Embedding from Language Models (ELMo). Given the problems of Word2vec and GloVe, words have different meanings in different contexts, and two words in the context of different model vector representations are the same. ELMo is optimized for this and can learn the complex characteristics of word usage and the changes in complex usage in different contexts.

3. Model

3.1. Model flow

This study uses the official estimates of ELMo and a training model for the word segmentation of domain names after the word vector transformation, resulting from the conversion of 3*8*1024 words vector data, where 3 is the output of a two-layer biLM with a layer of CNN for the output of character encoding; 8 is the length of the longest list, and a domain name can be divided into the longest length of a word group; and 1024 is the dimension of each layer of the output and input to the CNN model for training.

Domain name preprocessing, top-level domain (TLD) removal, and segmentation are included using Python's word segmentation toolkit, such as WordNinja. The number of words obtained is generally not more than 8. Thus, symbols can be used to supplement the insufficient number and unify the data length.

The CNN model used in this study was first proposed by Yoon Kim et al.[14]. CNN that is used for the text classification task is TextCNN. Given the textual data, after word segmentation and ELMo model, the domain name is processed as sentence matrix 8*1024; the convolutional layer uses one-dimensional convolution with 6 filters; after 1-max pooling layer, the softmax layer of the full connection layer and the probability of the categories are entered.

The specific model is shown in Fig.1.
In the above background, the detection algorithm based on ELMo and TextCNN in this paper is shown in Algorithm 1:

**Algorithm 1** Detection methods based on ELMo and TextCNN

1. Prepare the domain name data set
2. Remove the top-level domains from the data and use the WordNinja toolkit to divide the domains into word pairs, supplemented with commas for domains up to 8 words long
3. Use the officially trained ELMo model to carry out word vectorization processing for different word groups, so that 8*1024 word vector data can be obtained for each domain name
4. Word vector data is the embedding layer of TextCNN, and the embedding layer is an 8*1024 matrix
5. After one dimensional convolutional layer, there are two output channels for each kernel_size
6. Pass another pooling layer and output to the softmax layer to obtain the probability of whether the domain name is malicious or not

**END**

3.3. **Algorithm analysis**

For the method in this study, the ELMo model uses a trained model. Thus, the process of corpus training is not needed, but the process of word vector transformation still increases the experimental time.

Converting 10,000 phrases of length 8 using EMLo takes approximately 6 s. This process is insignificant for the training stage. However, the data dimension that needs to be processed in the training stage is increased from 1*75 (i.e., the data dimension of LSTM) and 6*128 (i.e., the data dimension of CNN) to 8*1024. This increase increases the time required for training, which is confirmed by subsequent experiments.

4. **Results**

4.1. **Experimental data**

Alexa ranking refers to a website’s world ranking, which provides multiple evaluation index information, including comprehensive ranking, visit volume ranking, and page view ranking. Most people regard Alexa ranking as the current authoritative website visit rating index. Therefore, the first 1 million domain name data of Alexa is selected as the legal domain name data set.

The data of malicious domain names are mainly collected from three typical malicious domain name families of Suppobox, Gozi, and Matsnu. The number of these three families generated every day is relatively small, and only 9,632 data samples are collected from the website of 360NetLab. However, in the later stage, the three-generation algorithms are used to supplement the samples, and approximately 100,000 samples of malicious domain names exist.

4.2. **Analysis of experimental results**
For the malicious domain names of different families, the difference lies in the generated algorithm. The malicious domain names of the same family have different dictionary seeds. Therefore, different experimental tests are set up, and the same or different dictionary seeds are used for the test and training sets.

In the first set of experiments, all legal domain names and malicious domain names, including 360NetLab data and samples generated from different dictionary seeds, were not subdivided. The ratio of the training, verification, and test sets is 8:1:1. The experimental results are shown in Tab.2.

| Model          | precision | F1   | TPR  | FPR  | Accuracy |
|----------------|-----------|------|------|------|----------|
| CNN            | 0.8998    | 0.9090 | 0.9185 | 0.1023 | 0.9081   |
| LSTM           | 0.9086    | 0.9164 | 0.9243 | 0.0930 | 0.9156   |
| CNN-LSTM       | 0.9155    | 0.9199 | 0.9243 | 0.0853 | 0.9195   |
| ELMo-TextCNN   | 0.9327    | 0.9428 | 0.9530 | 0.0687 | 0.9422   |

Tab.2 Mixed use of all samples and classification index data of each model

For the CNN model, after adding ELMo pre-trained word vector processing, the classification ability is improved. The word vector is helpful for the classification of typical malicious domain names.

The second experiment made a purposeful distinction between the sample data of the training and test sets such that the dictionary seeds used by some malicious domain name samples in the test set did not appear in the training set. The purpose is to test the generalization ability of each model. The experimental results are shown in Tab.3.

| Model          | precision | F1   | TPR  | FPR  | Accuracy |
|----------------|-----------|------|------|------|----------|
| CNN            | 0.7077    | 0.7800 | 0.8687 | 0.3587 | 0.7550   |
| LSTM           | 0.6766    | 0.7768 | 0.9120 | 0.4360 | 0.7380   |
| CNN-LSTM       | 0.7015    | 0.7890 | 0.9085 | 0.3865 | 0.7610   |
| ELMo-TextCNN   | 0.7743    | 0.8413 | 0.9210 | 0.2684 | 0.8263   |

Tab.3 Experiment 2, test set and training set use different dictionary seeds

The results show that, in the test of the generalization ability of the model, each model index presents a significant decline, among which the FPR value increases significantly. That is the error of classifying malicious domain names into legitimate domain names increases. Therefore, a complete training data set is needed to give full play to the model.

4.3. Time performance
For the first experiment, approximately 100,000 data sets, the training time, and the test time consumed by each model are shown in Tab.4.

| Model          | Training time(About 80000 data) | Test time(About 10000 data) |
|----------------|---------------------------------|-----------------------------|
| LSTM           | 24min                           | 12s                         |
| CNN            | 28min                           | 8s                          |
| CNN-LSTM       | 34min                           | 14s                         |
The ELMo-TextCNN model used in this study takes the longest time, which is also predictable, because the pre-trained word vector model is used, and the word dimension is much larger than the previous model. Thus, training time increases. For the testing stage, the process of domain name preprocessing and ELMo transformation of the word vector makes the testing time slightly longer than the previous model.

5. Conclusion
DGA detection, which is the classification task of distinguishing benign domain names from malicious code generation domain names, has become an important topic in information security research. This study concludes that, when the data samples are abundant, the model in this study can achieve the best on multiple indexes, but the FPR value remains low. The model has obvious advantages for the random forest method of manual extraction of features. However, in the actual situation, we still need to face problems, such as insufficient data volume and unbalanced data set. Moreover, the typical malicious domain name is only one of the difficulties in the detection of malicious domain names.

6. Acknowledgments
This work is supported by the National Natural Science Foundation of China (NSFC) under Grant 61972187, the Scientific Research Project of Science and Education Park Development Center of Fuzhou University, Jinjiang under Grant 2019-JJFDKY-53.

References
[1] Daniel P, Khaled Y, Michael K, Johannes B and Elmar G 2016 A Comprehensive Measurement Study of Domain Generating Malware Proc. 25th {USENIX} Security Symposium (Aug 16 Vancouver, BC, Canada) pp 263-278
[2] Bin Y, Jie P, Jiaming H, Anderson N and Martine D 2018 Character level based detection of DGA domain names. Proc. Int. Joint Conf. on Neural Networks (Jul 8, Rio, Brazil) pp 1-8
[3] Yoshua B, Réjean D, Pascal V and Christian J A neural probabilistic language model. 2003 Journal of machine learning research pp 115-1137
[4] Jason G 2013 End-to-end analysis of a domain generating algorithm malware family Black Hat USA
[5] BitdefenderLabs 2014 Tracking Rovnix V Available Online: Tracking Rovnix. http://labs.bitdefender.com/2014/11/ tracking-rovnix-2/
[6] Stanislav S 2015 MATSNU Technical Report
[7] Schüppen S, Teubert D, Herrmann P and Meyer U 2018 {FANCI } Feature-based automated domain classification and intelligence. Proc. 27th {USENIX} Security Symposium(Aug 15, Baltimore, USA ) pp1165-1181
[8] Anderson Hyrum S, Woodbridge J and Filar B DeepDG 2018 A Adversarially-tuned domain generation and detection Proc. the 2016 ACM Workshop on Artificial Intelligence and Security (Oct 28, Hofburg, Austria) pp 13-21
[9] Tran D, Mac H, Tong V, Tran A and Nguyen G A LSTM based framework for handling multiclass imbalance in DGA botnet detection Neurocomputing 275 pp 2401-2413
[10] Pereira M, Coleman S, Yu B, DeCock M and Nascimento A 2018 Dictionary extraction and detection of algorithmically generated domain names in passive DNS traffic Proc. the 21st Int. Symposium on Research in Attacks (Heraclion, Greece) pp 295-314
[11] Kim Y 2014 Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882
[12] Mikolov T, Chen K, Corrado G and Dean J 2013 Efficient estimation of word representations in vector space arXiv preprint arXiv:1301.3781
[13] Pennington J, Socher R and Manning D 2014 Glove: Global vectors for word representation. *Proc. of the 2014 Conf. on empirical methods in natural language processing* (Oct 29, Doha, Qatar) pp1532-1543

[14] Zhang Y and Wallace B 2015 A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification *arXiv preprint arXiv:1510.03820*