Detecting Dictionary Based AGDs Based on Community Detection

Qianying Shen and Futai Zou(✉)

Shanghai Jiaotong University, Shanghai, China
{sjtusqy,zft}@sjtu.edu.cn

Abstract. Domain generation algorithms (DGA) are widely used by malware families to realize remote control. Researchers have tried to adopt deep learning methods to detect algorithmically generated domains (AGD) automatically based on only domain strings alone. Usually, such methods analyze the structure and semantic features of domain strings since simple AGDs show great difference in these two aspects. Among various types of AGDs, dictionary-based AGDs are unique for its semantic similarity to normal domains, which makes such detections based on only domain strings difficult. In this paper, we observe that the relationship between domains generated based on a same dictionary shows graphical features. We focus on the detection of dictionary-based AGDs and proposes Word-Map which is based on community detection algorithm to detect dictionary-based AGDs. Word-Map achieved an accuracy above 98.5% and recall rate above 99.0% on testing sets.

Keywords: Algorithmically generated domains · Community detection · Machine learning

1 Introduction

Algorithmically generated domains (AGDs) refer to a group of domains generated in batches based a string of random seeds [1]. According to different generation algorithms, AGDs can be roughly divided into four categories: arithmetic based, hashing based, permutation based and word dictionary based [2]. The first three types of AGDS are often in forms of a random combination of letters and numbers, which is obviously different from the normal domain names in aspects of lexical and semantic characteristics. Dictionary based AGDs discussed in this paper are generated from a random combination of commonly used English words, the lexical and semantic characteristics of which show little difference with normal domains.

In this paper, a new method named Word-Map is proposed to solve the problem that dictionary-based AGDs are difficult to detect using lexical and semantic characteristics. Word-Map is designed to achieve two effects: actively mine DGA dictionaries and

This work is supported by the National Key Research and Development Program of China (No.2017YFB0802300).

N. Park et al. (Eds.): SecureComm 2020, LNICST 335, pp. 42–48, 2020.
https://doi.org/10.1007/978-3-030-63086-7_3

accurately detect dictionary-based AGDs. The key idea of Word-Map is to convert the problem of dictionary-based AGDs detection into a community detection problem on a word map which is constructed based on the co-occurrence of words in a certain set of domains. Data used as training set and test set in this paper are composed of Suppobox domains from DGArchive dataset and Alexa Top 1 M domains. Suppobox domains comes from three different dictionaries. Word-Map has achieved good results in testing set. The accuracy and recall rate of Word-map on domains from same DGA dictionaries, domains from different DGA dictionaries and imbalanced dataset is respectively above 98.5% and 99.0%.

2 Methodology

2.1 Word Graph

Word Graph Construction

For a domain string set D = {d1, d2,……, di,……, dn}, we process each domain name by removing the top-level domain string, and only retaining the second-level domain string. Then we get a second-level domain string set S = {s1, s2,……, si,…, sn}. Then we cut each element in S into single English words. A mature tool named Wordninja [3] can be used to solve this problem efficiently. This paper uses Wordninjia to cut the domain strings into words.

For a domain name set D, after domain string splitting, a word set W can be obtained, and each word in the word set is a vertex. These word vertexes will then be connected by edges according to their co-occurrence relationship in D. In this way, for a domain name set D, a word graph G = (V, E) can be obtained.

An example of the composition process is shown below:

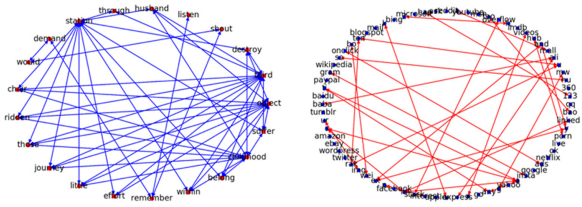

Fig. 1. Word graph of dictionary based AGDs and Alexa domains

Fifty domain strings are respectively randomly chosen from the dictionary based AGDs dataset which are from the same dictionary and Alexa domains dataset. As shown in Fig. 1, there are 21 word nodes and 100 directed edges in DGA-Graph, while there are 72 word nodes and 60 directed edges in Alexa-graph. We can see that the average degree of vertexes in Alexa-graph is less than AGD-Graph. It's because some domains is too short to be cut, therefore becoming isolated vertex in the word graph. It can be intuitively observed that for equivalent amount of dictionary based AGDs and Alexa domains, dictionary based AGDs will be cut into fewer word vertexes, however dictionary based

AGDs word vertexes of are more closely connected, and the average degree of vertexes is higher. This shows that the word vertexes obtained from dictionary based AGDs are more closely related, which makes them easier to be classified into a community

2.2 Community Detection on Word Graph

Introduction of Infomap

The key idea of Word-Map is to convert the detection of dictionary-based AGDs into a community detection problem on word graphs. The key step is to use the Infomap [4] algorithm to perform community detection on word graphs.

Suppose there are M vertexes, which are divided into m communities Infomap. Formula (2) [5] describes the average code length of each step of a random walk on the graph after community division.$q_{\curvearrowright}$ represents the proportion of community codes in all codes. $H(\mathcal{Q})$ represents the average length of community codes, and $P^i_{\circlearrowleft}$ represents the proportion of vertexes codes and leaving action codes belonging to community i in all codes.$H\left(\mathcal{P}^i\right)$ represents the average length of all vertex codes and leaving action codes of community i. L(M) is the average length of each step of random walk on the graph after community division. Obviously, obtaining the best code is equivalent to minimizing L(M).

$$L(M) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} P^i_{\circlearrowleft} H\left(\mathcal{P}^i\right) \tag{1}$$

Words Community

Following figures are examples of community detection results of word graphs obtained from dictionary based AGDs, Alexa domains, and mixed domains.

Fig. 2. Communities detected from Alexa-graph

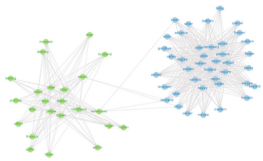

Fig. 3. Communities detected from AGD-graph

As shown in Fig. 2, 46 communities were detected on the word graph obtained from 50 Alexa domains as shown in Fig. 2. The largest community is composed of 6 blue colored vertexes in the center of Fig. 4, of which the total degrees is 14 and the average degree is 2.33. For other communities, The number of vertexes is 1 (isolated vertex), 2 (word vertexes obtained from a same domain string) or 3. It can be seen that words obtained by splitting Alexa domains have no obvious clustering characteristics.

As shown in Fig. 3, 2 communities were detected on the word graph obtained from 50 dictionary based AGDs as shown in Fig. 1. The two communities are composed of 40 and 24 vertexes respectively, of which the total degrees is 739 and 261 respectively and the average degree is 18.48 and 10.88 respectively. Compared to Fig. 2, it can be seen that words obtained by splitting dictionary based AGDs have obvious clustering characteristic. Also, there are obvious differences in aspects of the number of vertexes, the total degrees of vertexes, and the average degree of vertexes between AGD communities and Alexa communities.

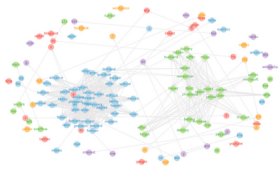

Fig. 4. Communities detected from the word graph of mixed domains. (Color figure online)

Mix the 50 Alexa domains and 50 dictionary based AGDs mentioned before and get the word graph of the mixed domains. 48 communities were detected on the word graph. The biggest two communities(blue and green colored vertexes in the center of Fig. 4) have 40 and 24 vertexes respectively. Their total degrees are 739 and 261, and their average degrees are 18.48 and 10.88. The number of vertexes of the remaining communities are all no bigger than 6, and their total degrees are no bigger than 14,while average degrees no bigger than 2.5. Comparing Fig. 4 with Fig. 2 and Fig. 3, it can be seen that dictionary based AGDs and Alexa domains have been effectively distinguished from each other.

It can be seen intuitively that there are the several differences between dictionary based AGD community and Alexa community: 1) The number of vertexes in dictionary based AGD communities is much bigger than that of Alexa communities. 2) The total and average degrees of the vertexes of dictionary based AGD are much bigger than those Alexa communities. Therefore, after community detection, these two features can be collected to train a decision tree to classify whether a community is dictionary based AGD community or not (Fig. 5).

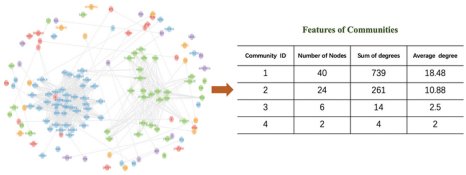

Fig. 5. Extract features of word Communities

3 Experiments and Results

3.1 Dataset

Dataset used in the experiments contains a total of 1,313,571 dictionary based AGDs domains and 1,000,000 benign domains. Dictionary based AGDs come from the Suppobox domains in the DGArchive dataset [6], and benign domains are chosen from Alexa Top 1 M domains.

Ground Truth Data As shown in Table 1., we randomly select 50,000 Alexa domains and 150,000 dictionary based AGDs to make up the training set, where the dictionary based AGDs are composed of words from three different dictionaries named D1, D2, and D3. We randomly select 50,000 Alexa domains and 150,000 dictionary based AGDs from the rest data to make up the testing set, where the dictionary based AGDs are also composed of words from three different dictionaries named D1, D2, and D3.

3.2 Experiments

We designed three different experiments to verify the performance of Word-map on domains from the same DGA dictionary, domains from different DGA dictionaries and imbalanced data sets.

The first group of experiments verifies Word-map's performance on domains from the same DGA dictionary with 3 round independent experiments as shown in Table 1.

Table 1. Experiment on domains from same DGA dictionaries

Dataset	Training set				Testing set			
	Alexa	D1	D2	D3	Alexa	D1	D2	D3
Round 1	50,000	50,000	0	0	50,000	50,000	0	0
Round 2	50,000	0	50,000	0	50,000	0	50,000	0
Round 3	50,000	0	0	50,000	50,000	0	0	50,000

The results of three independent experiments are shown in Table 2.

Table 2. Performance on domains from same DGA dictionaries

	Round 1	Round 2	Round 3
Accuracy	99.7%	99.5%	99.8%
Recall rate	100.0%	99.7%	99.9%

The second group of experiments verifies Word-map's performance on domains from different DGA dictionaries with 3 round independent experiments (Table 3).

Table 3. Experiment on domains from different DGA dictionaries

Dataset	Training set				Testing set			
	Alexa	D1	D2	D3	Alexa	D1	D2	D3
Round 1	50,000	50,000	50,000	0	50,000	0	0	50,000
Round 2	50,000	50,000	0	50,000	50,000	0	50,000	0
Round 3	50,000	0	50,000	50,000	50,000	50,000	0	0

The results of three independent experiments are shown in Table 4.

Table 4. Performance on domains from different DGA dictionaries

	Round 1	Round 2	Round 3
Accuracy	99.3%	99.4%	99.1%
Recall rate	99.7%	99.9%	99.6%

The third group of experiments verifies the performance of Word-map on imbalanced dataset with 3 round of independent experiments as shown in Table 5.

Table 5. Experiment on imbalanced dataset

Dataset	Training set				Testing set			
	Alexa	D1	D2	D3	Alexa	D1	D2	D3
Round 1	50,000	500	500	500	50,000	500	500	500
Round 2	50,000	500	500	0	50,000	0	0	500
Round 3	50,000	500	0	0	50,000	500	0	0

The results of three independent experiments are shown in Table 6.

Table 6. Performance on imbalanced dataset

	Round 1	Round 2	Round 3
Accuracy	98.5%	98.8%	98.6%
Recall rate	99.2%	99.0%	99.1%

Based on the results of the three rounds of experiments, it can be seen that when DGA dictionaries are known in advance, Word-map can accurately detect dictionary based AGDs. The performance of Word-map is not that good when AGDs are from different dictionaries, but its accuracy and recall rates still remains above 99%, indicating that Word-map has sufficient ability to mine new dictionaries. As to imbalanced data sets, the accuracy and recall rate of Word-map are also kept above 98.5% and 99.0% respectively, indicating that Word-map can adapt well to the actual scenarios where the number of dictionary based AGDs is far small than the number of benign domain names.

References

1. Plohmann, D., Yakdan, K., Klatt, M., et al.: A comprehensive measurement study of domain generating malware. In: 25th {USENIX} Security Symposium ({USENIX} Security 16), pp. 263–278 (2016)
2. Sood, A.K., Zeadally, S.: A taxonomy of domain-generation algorithms. IEEE Secur. Priv. **14**(4), 46–53 (2016)
3. wordninja, https://github.com/keredson/wordninja
4. Rosvall, M., Bergstrom, C.T.: Maps of information flow reveal community structure in complex networks. arXiv preprint physics.soc-ph/0707.0609 (2007)
5. Rosvall, M., Axelsson, D., Bergstrom, C.T.: The map equation. Eur. Phys. J. Spec. Topics **178**(1), 13–23 (2009)
6. Plohmann, D., Yakdan, K., Klatt, M., Bader, J., Gerhards-Padilla, E.: A comprehensive measurement study of domain generating malware. In: 25th USENIX Security Symposium, pp. 263–278 (2016)