International Conference on Mathematics, Modelling, Simulation and Algorithms (MMSA 2018)

A Detection Scheme for DGA Domain Names Based on SVM

Zhen Wang[1,2], Zhongtian Jia[1,2,*] and Bo Zhang[1,2]

[1]School of Information Science and Engineering, University of Jinan, Jinan 250022, China

[2]Provincial Key Laboratory for Network Based Intelligent Computing, University of Jinan, Jinan 250022, China

[*]Corresponding author

***Abstract*—Most of network security configurations allow the DNS data to pass through. Therefore, the crackers often embed malware commands in DNS data to avoid the security detection by the Internet facilities. Especially, some malwares, such as the botnet, generate a large number of spare domain names using a Domain Generation Algorithm (DGA) and choose some of them as the masks of malware's commands. How to filter out the DGA domain names from the normal domain names becomes a hot topic in literature. There are many papers trying to solve this problem. However, the comprehensive analysis of the character features of the domain name is absent. In this paper, we studied the characters' features of DGA domain names and extracted five attributes for the Support Vector Machine (SVM) model. Model training and cross-validation showed that the detecting accuracy, the precision, and the recall rate were greater than 91%, 88%, and 87%, respectively. Experiments also illustrated that compared with the decision-tree method, the detecting algorithm based on SVM could obtain higher accuracy, precision and recall rate.**

Keywords—DNS; domain name; DGA; SVM; decision-tree

I. INTRODUCTION

The Domain Name System (DNS), on which an overwhelming majority of Internet applications depends, is an important Internet infrastructure. Therefore, most of network security configurations allow the DNS data to pass through. It arouses a great interest to the malicious attackers. They often embed malware commands in DNS data to avoid the security detection by the Internet facilities. In order to do so, they need a Domain Generation Algorithm (DGA) to produce a large number of peculiar domain names and choose some of the domain names register at random. For example, the malware conficker [1-2] and srizbi [3] are both the typical use of this technology. In botnet research, Gu et al. [4] proposed a system to study botnets. The system identifies the botnet by clustering the malicious traffic without the restricting of the Command and Control channels (C&C) protocol and the botnet structure. In addition, it does not need the prior knowledge of the botnet. However, some malwares, such as the botnet, employ the DGA to generate a great number of spare domain names. This leads to high overhead and difficulty in the detection of the malicious domain names. It is impossible to use a blacklist to block a malicious domain name later. Therefore, the academia begins to look for new ways to detect the malicious domain names.

The earliest research on DGA domain names used the reverse and honeypot technology. By reverse engineering, it was possible to master the domain generation algorithm and domain usage mechanism used by malicious samples. Stone-Gross et al. [5] analyzed the Torpig samples and found the domain generation algorithm. Hence, they successfully controlled the botnet about 10 days by preempting domain names and deploying a fake C&C server. However, the reverse engineering costs many time, labor and financial resources. In recent years, people have tried to detect DGA domain names from the linguistic characteristics of strings and analyzing traffic data. Through a lot of observation and experiment, McGrath et al. [6] found that normal URLs (Uniform Resource Locator) and malicious URLs exhibit different string language characteristics. In [7], Ma et al. analyzed the string characteristics of the URLs to judge whether a domain name was malicious or not. The string characteristics included the domain name length, the host name, the number of URLs, and the characters of a URL. Stalmans et al. [8] designed a system to detect the rapidly changing domain names by observing the query traffic of domain names. In the system, they analyzed the extracted features and detected the malicious domain names with the decision-tree algorithm of C5.0 version and the Bayesian statistical learning method.

In 2010, by capturing the client's domain resolution requests and response records, Yadav et al. [9] analyzed the distribution characteristics of the characters in domain names and found that the differences of characters' distribution between the normal domain names and the DGA domain names. Taking the domain KL distance, edit distance and Jaccard correlation coefficient as feature vector, they identified the DGA domain names with machine learning method. In 2011, Yadav et al. [10] detected C&C server addresses and botnets by monitoring NXDOMAIN (the failed domain queries) traffic and the information entropy of domain names. In the same year, Antonakakis et al. [11] inspected the randomly generated domain names using the traffic data that did not contain any domain names. They believed that there must be traffic data that did not embody the domain names in the same Botnet. Therefore, by collecting analyzing a large number of such traffic data, they distinguished the malicious domain names with clustering and classification method. In 2013, Zhou et al. [12] observed the DGA domain names by filtering NXDOMAIN traffics. Because the DGA names had the similar life times and query modes, they could effectively filtered out the DGA domain names. In 2014, Schiavoni et al.

[13] used the Phoenix mechanism to find out the DGA domain names. By the analysis of the domain string features and IP features, they figured out the domain of the botnet. In 2016, by checking DNS traffics and research the linguistic features of domain names, Tong et al. [14] classified normal domain names and DGA domain names.

To the detecting methods based on the characteristics of network traffic, it is often necessary to obtain a large amount of domains resolution data from a top-level domain name server or a recursive resolution server. Nevertheless, whether from the top-level domain server or from the recursive analytical server, it is both very difficult to obtain the traffic data because this is a time-consuming work. Therefore, from the structure and the character distribution to analyze the DGA domain names and design the detection scheme have gradually become a research hotspot in the field in literature. Davuth et al. [15] analyzed the characters of the domain names. However, they only used the accuracy to measure the model and did not evaluate the model in depth. In order to solve this problem, we use the Support Vector Machine (SVM) [16-17] algorithm to study the characters' distribution of DGA domain names. In experiments, we take the open DGA domain data set samples. Through practical observations and many experiments, we draw out five features of domain names. Experiments and training results shows that the extracted attributes is adequate for the detection. The proposed scheme has three characteristics. 1) The data set updates every day and is easily to get, which guarantee validity and accuracy of our experiments; 2) To obtain a high accuracy, we focuses on the detecting of DGA domain names regardless of the other type of malicious domains; 3) To improve the efficiency, we only study the domain strings ignoring the other attributes deliberately. The contributions of the paper are threefold. 1) We extract five features of the domain names using statistical method; 2) We work out the detecting accuracy, precision, and TPR by model training and cross-validation; 3) We compared the experiment results gotten from the SVM algorithm and the decision-tree method.

The reminder parts of this paper are organized as follows. In section II, we propose a detection scheme for DGA domain names. Section III presents the experimental verification and analysis of the experimental results. Section IV compares the experiment results getting from the SVM algorithm and the decision-tree method. Conclusions are drawn in section V.

II. PROPERTIES ANALYSIS

There are the two significant differences between the normal domain name and a DGA domain name. One is the organic component of the domain name; the other is the character's distribution of the domain name. In detail, a normal domain name usually consists of English words or their abbreviations, while a DGA domain name is often composed of clustered letters and random numbers. The features of a normal domain name are: 1)the length of domain name is short; 2)the name's composition and the character's distribution both have some rules to follow; 3)a clear implication exists in a normal domain name; and 4)a normal domain name has strong readability suitable for the reading demands. Relatively a DGA domain name is featured in: 1) the length of a DGA domain name is particularly long；2) the name's composition and the character's distribution are both chaotic; 3) there is not a clear meaning in a DGA domain name; and 4) a DGA domain name is not readable. In the remainder part of this section, we will study the attributes of the DGA domain name in detail.

A. Domain Name Length

TABLE I. A CASE OF LARGE DIFFERENCE IN LENGTH

Normal Domain Name(Len)	DGA Domain Name (Len)
google.com(9)	hjbtestnessbiophysicalohax.com(29)
facebook.com(11)	kwtoestnessbiophysicalohax.com (29)
youtube.com(10)	earnestnessbiophysicalohax.com(29)
yahoo.com(8)	txmoestnessbiophysicalohax.com (29)

TABLE II. A CASE OF LITTLE DIFFERENCE IN LENGTH

Normal Domain Name (Len)	DGA Domain Name (Len)
google.com(9)	qldut.com(8)
facebook.com(11)	icdnt.org(8)
youtube.com(10)	otvovlf.com(10)
yahoo.com(8)	lnhco.info(9)

In order to avoid duplication with the registered domain names, the length of a DGA domain name is usually longer than that of a normal domain name. As shown in Table I, a normal domain name's length is generally shorter than 20, while the length of a DGA domain name is usually longer than that. The DGA domain names listed out in Table I, which are from the Banjiri malware [18], are only as an example in the paper. In fact, the length of some other DGA domain names may be even longer than the length of the names shown in Table I. However, it is impossible to decide whether a domain name is normal or not only according to the domain name length. Because of the fact that some DGA domain names are very short at times. As Illustrated in Table II, the normal domain name and the DGA domain name, which is from the malware Conficker [19], have little differences in the length. Therefore, in order to identify the DGA domain names, more features should be survey, such as the chaotic extent of the DGA domain name.

B. Domain Name Entropy

In this subsection, we discuss the entropy of a DGA domain name. The size of the entropy [20] indicates the chaotic degree of a system. The greater the entropy, the more chaotic the system is. As far as the domain name is concerned, a DGA domain name's character distribution is more stochastic, and its' chaos degree is far greater, compared with those of the normal domain name. Hence, the entropy of a DGA name should be larger than that of a normal domain name. To prove this inference, we take 2000 normal domain names and 2000 DGA domain names used by a malware, such as Conficker and Gameover, as a sample and calculate their entropy respectively. Figure I shows the calculation results. From Figure I, we can see that the entropy of a DGA name is significantly larger than that of the normal domain name indeed. Hence, the domain name's entropy should be another factor of DGA name's research.

Note: We use the formula (1) to calculate the entropy of domain name, where X is a random variable, X= {X1, …, Xn}. Here the parameter b is the base of the logarithm and b=2. Bit is unit of the entropy.

$$H(X) = \sum_i P(X_i) \, log_b P(X_i) \qquad (1)$$

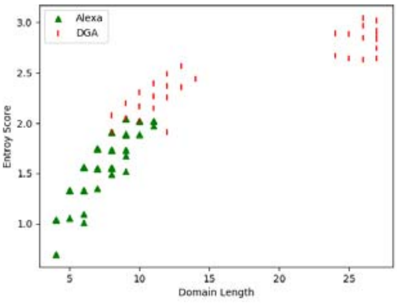

FIGURE I. ENTROPY OF DOMAIN NAMES

C. *Vowel's Ratio in a Domain Name*

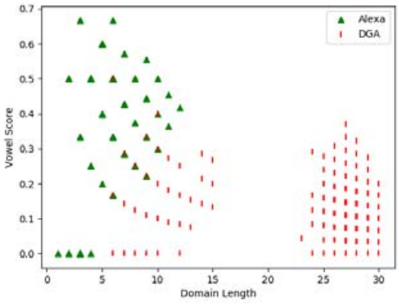

FIGURE II. VOWELS' RATIO IN DOMAIN NAMES

In order to remember conveniently, people usually adopts some easily readable words or their abbrreviations domain name, such as google, yahoo, anta, etc. Obviously, the domain names are all composed of consonants and vowels and the two type of letters' arrangement is very logical, which make the domain names readable. In comparison, a DGA domain name generates randomly by some letters and numbers, which cause the name hard to read. Take fryjntzfvti. biz as an example, it hard to find any arraying rule and any readability in the name. Especially, we note that the vowels' proportion acts as a key factor in the composition of a normal domain name. Higher vowels' ratio not only makes a normal domain name looks more methodical, but also bring better readability. Figure II gives out the result of the statistics of 2000 normal domain names and 2000 DGA domain names used by malware Conficker and Gameover. From Figure II, we find that the proportions of vowels in the normal domain name and the DGA domain name are very different. Generally, no matter how long the DGA domain name is, it contains significantly fewer vowel letters than the normal domain names. Low proportion of vowels is only special case, in which the domain name registrants hope their domain names imply some particular meaning, such as 360.cn, 51job.com. Therefore, the vowel's Ratio in Domain Names must be a factor of the DGA name investigation.

D. *Consecutive Consonants' Ratio in a Domain Name*

Large number of observations and experiments illustrate that the consecutive consonants's proportion of DGA domain names is greater than that of normal domain names. Figure III shows the statistics of 2000 normal domain names and 2000 DGA domain names used by the malware Conficker and Gameover. From Figure III, we discover that the logner a domain name is, the higher proportion the continuous consonants has. Compared with the normal domain names, the continuous consonants in DGA domain names, whose length exceeds 20, have a higher proportion. Under the same length condition, the proportion of consecutive consonants in a DGA domain name is obviously higher than that in a normal domain name. Therefore, the proportion of consecutive consonants in domain names should become a factor to research the DGA domain names.

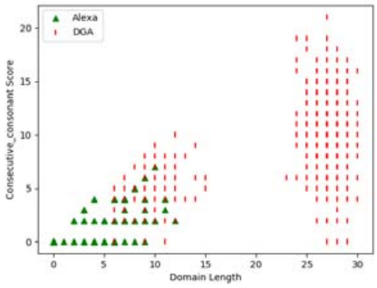

FIGURE III. PROPORTION OF CONSECUTIVE CONSONANTS IN DOMAIN NAMES

E. *Proportion of the Digits in a Domain Name*

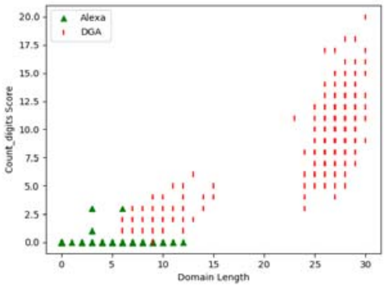

FIGURE IV. DIGITS PROPORTION IN DOMAIN NAMES

The normal domain name, except a very little number of domain names, seldom contains numbers; while the DGA domain name's composition are usually inseparable from digits. Figure IV is a statistics result of 2000 normal domain names and 2000 DGA domain names used by malware Conficker and Gameover. From Figure IV, we can see that DGA names contain a higher proportion of numbers, and the proportions of the numbers in the normal domain names are not significantly different when the length of the domain names is changed. Generally, a normal domain name may have some meaning in some semantic language environments, such as facebook.com, taobao.com and so on. However, no matter in what kind of language environment, the DGA domain name's semantics are not obvious. Therefore, the proportion of the digits should be the fifth factor to detect the malicious domain names.

III. Experiment Analysis

A. Data Set of the Experiment

The data set of normal domain names is from www.alexa.com, which provides traffic ranks of the worldwide web over the past three months. Top domain names are generally normal domain names. In the experiment, we use the domain names of top ten thousand in Alexa website as the normal domain names. The data set of the DGA domain names comes from the website data.netlab.360.com/dga/, which updates the DGA domain name every day. In the experiments, we mix up 2000 normal domain names with 2000 DGA domain names and take them as experiment samples.

B. Label Domain Names

After we get the experiment samples, we need to label the domain names. In the experiment, we label the normal domain name with the flag '0' and tag the DGA domain name with the flag '1'.

C. Feature Selection and Feature Vector's Extraction

Feature selection has a direct impact on the model's training and directly relate to the success or failure of the model. Therefore, features selection is the key to this paper. If we select too many features, not only the trained model is easily over-fitting, but also the generalization ability of the model will be weaken. However, if we select too few features, the fitting degree of the model will be affected and the training accuracy of the model will be reduced. According to the discussion of the second part of this paper, we select the following five features: 1) domain name length; 2) domain names entropy; 3); 4) consecutive consonants' ratio in domain names; and 5) proportion of the digits in domain names. Therefore, the feature vector is expressed by the variable X, $X = (l, e, r_1, r_2, r_3)^T$, in which l represents the domain name length, e indicates the the domain name entropy, r_1 acts as vowels' ratio in domain names, r_2 is the consecutive consonants' ratio in domain names, and r_3 serves as proportion of the digits in domain names.

D. Feature Processing

In order to improve the computation speed and process the samples conveniently, we normalize the characteristic values. In experiments, we used the normalization method of Z-score. The conversion function is in formula (2).

$$\bar{x} = \frac{x-\varepsilon}{\delta} \qquad (2)$$

Where ε represents the mean value of a characteristics, δ works as the standard deviation of a characteristics value, and *x* is the value of a given characteristics.

E. Algorithm Selection

In experiments, we use the SVM as the experimental classification algorithm. SVM is a linear classification algorithm based on structure risk minimization [21] and VC dimension theory [22]. It is a machine-learning algorithm of artificial intelligence. By finding the largest classification hyperplane in the feature space of sample, this algorithm can achieve better classification effect. This classification method uses the hyperplane in the eigenspace to enable the learner get global optimization. In addition, it can make the expectation risk of the whole sample space obtain an upper bound of probability. SVM algorithm has rigorous mathematical derivation and theoretical basis, and it has a high accuracy in classification. Further, it has strong stability and high generalization ability. Because the support vector decides the result of the SVM algorithm, we can seize the samples of the key support vectors and discard the samples with small correlation when we adopt SVM classification algorithm to reduce the complexity of computation. There are usually two situations when we use the SVM classification algorithm. One is a linear separable case, and the other is a nonlinear separable case. For the five features selected in this paper, the distinguishing between a normal domain name and a DGA domain name belongs to the linear separatable case [23]. Therefore, a hyperplane that can separate the training samples must exist when we use the SVM algorithm. The hyperplane takes the form as formula (3).

$$g(X) = WX + b \qquad (3)$$

Where X is the feature vector, W is the weight vector, and b is the intercept, also known as the threshold weight or offset. If $g(X) > 0$, then the samples are above the decision plane, and if $g(X)$<0, the samples are under the decision plane.

F. Cross Validation

Our experiment belongs to the binary classification, and the classification results in the binary classification will appear in the following four situations, as shown in Table 3.

TABLE III. POSSIBLE PREDICTION RESULT IN BINARY CLASSIFICATION

Actual sample	Prediction to Positive	Prediction to Negative
Positive Sample	True Positive (*TP*)	False Negative (*FN*)
Negative Sample	False Positive (*FP*)	True Negative (*TN*)

In Table III, TP indicates that we predict the positive samples to positive terms, FP means that the negative samples are predicted to be positive terms, FN is the case that the positive samples are believed as negative terms, and TN represents that the negative samples are treated as negative terms. In experiments, we take DGA domain names as positive samples and normal domain names as negative samples, respectively.

(a) If the classifier can correctly detect the domain names generated by DGA, that is, the positive samples are predicted into positive class, set TP = 1;

(b) If the classifier correctly detects the normal domain name, that is, the negative samples are detected as negative class, set TN = 1

(c) If the classifier can not correctly detect the domain names generated by DGA, that is, the positive samples are predicted to negative class, then set FN = 1;

(d) If the classifier can not correctly detect the normal domain name, that is, the negative samples are predicted to positive terms, then set FP = 1.

In our experiments, we use "Precision", "TPR" and "Accuracy" as performance merits of the model. Precision is the ratio of TP to the sum of TP and FP. Formula (4) illustrates the Precision's calculation method.

$$Precision = \frac{TP}{(TP+FP)} \tag{4}$$

TPR is also called "Recall Rate", it is the ratio of TP to the sum of TP and FN. The TPR's computing method is as in formula (5).

$$TPR = \frac{TP}{(TP + FN)} \tag{5}$$

As shown in formula (6), accuracy represents the ratio of the sum of TP and TN to the sum of TP, FN, FP, and TN.

$$Accuracy = \frac{TP+TN}{(TP+FN+FP+TN)} \tag{6}$$

In experiments, we first select 4000 normal domain names and 6000 DGA domain names. Then we combine them together randomly. Finally, we choose 30% of the samples as a validation set and 70% of samples as a training set.

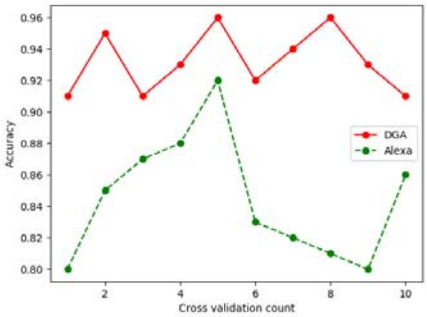

FIGURE V. RELATIONSHIPS BETWEEN THE CROSS-VALIDATION AND ACCURACY

The validation set includes 1000 normal domain names and 2000 DGA domain names. Through 10 times cross validating [24], we obtained the best values for the parameters, c = 9.0, and g = 0.6, where g is the kernel parameter and c is the penalty factor. Figure V and Figure VI show the experiment results.

From Figure V, we find that the accuracy DGA domain names in each time classification exceeds 91% and the accuracy of normal domain names in each time classification is over 81% in each cross-validation.

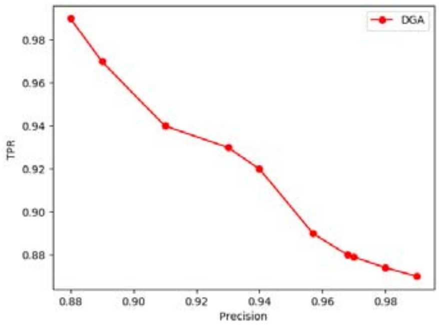

FIGURE VI. RELATIONSHIP BETWEEN PRECISION AND TPR

As can be seen from Figure VI, the TPRs in the experiment are more than 87%, and the precisions are above 88%. Importantly, in a certain experiment TP reaches 95.3%, TN gets to 93.4%, FP is low to 6.6%, and FN is 4.7%.

IV. COMPARISON BETWEEN DIFFERENT METHODS

We use the decision-tree to do experiment using the same data set and the same characteristic values. Then we compare the results of the experiment with those of the SVM method. The comparison of "Precision" and "TPR" in the two classification methods are shown in Figure VII.

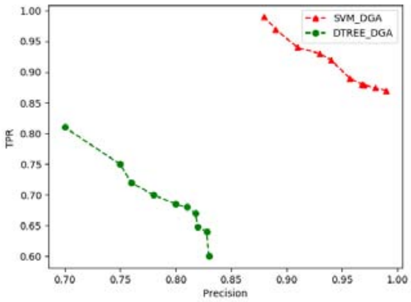

FIGURE VII. VII PRECISION AND TPR IN THE TWO CLASSIFICATION METHODS

TABLE IV. COMPARISON BETWEEN DIFFERENT CLASSIFICAITON METHODS

Methods	TP	TN	FP	FN
SVM	95.3%	93.4%	6.6%	4.7%
Decision-Tree	85.2%	79.2%	10.8%	14.8%

In the Figure VII, we can see that the highest precision is lower than 83.2% and the highest TPR is no more than 81.2% in the classification decision-tree method. In a certain experiment, if the classification decision-tree is adopted, TP, TN, FP and FN are 85.2%, 79.2%, 10.8% and 14.8, respectively. The comparison is as in Table IV.

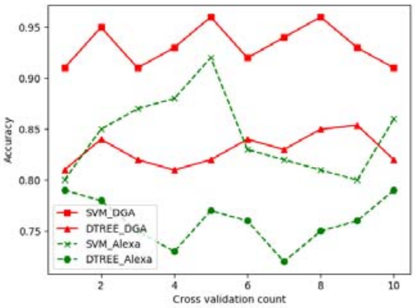

FIGURE VIII. CROSS-VALIDATION IN THE TWO CLASSIFICATION METHODS

Figure VIII list out the comparison result in each cross-validation between the SVM method and the decision-tree algorithm. From Figure VIII, detection for the SVM domain names or checking for the normal domain names, the accuracy gotten from the SVM method is higher than that is from the decision-tree way in each cross-validation.

V. CONCLUSIONS

In the paper, we propose a detection scheme for the DGA domain names using SVM algorithm. In the scheme, we select five typical features of the DGA domain names through data analysis and statistical method. The five selected attributes are 1) domain name length, 2) domain name entropy, 3) vowel's ratio in a domain name, 4) consecutive consonants' ratio in a domain name, 5) proportion of the digits in a domain name. These five features constitute a tuple of five elements. By normalizing the characteristic values with the normalization of Z-score, we improve the computation speed. In each experiment, the TPR is greater than 87% and the precision is above 88%. Importantly, in experiments TP, TN are greater than 95.3%, 93.4%, respectively. The FP, FN are smaller than 6.6%, and 4.7%, respectively. Compared with the other classification method, such as the decision-tree, the SVM can obtain better experiment result.

However, the limitation of this paper is that the data set of normal domain names and DGA domain names is relatively small, which has an adverse impact on the accuracy of the model. Moreover, some features selected by the model cause a large amount of computation. In future work, we will pursue this study and get a better detection model.

ACKNOWLEDGEMENT

This work is supported by the Natural Science Foundation of Shandong Province (No. ZR2013FL003, ZR2014FL011).

REFERENCES

[1] Fitzgibbon N, Wood M. Conficker. C: A technical analysis [J]. SophosLabs, Sophos Inc, 2009.

[2] Shin S, Gu G. Conficker and beyond: a large-scale empirical study [C]//Proceedings of the 26th Annual Computer Security Applications Conference. ACM, 2010: 151-160.

[3] Krebs B. Takedowns: The shuns and stuns that take the fight to the enemy [J]. McAfee Security Journal, 2010, 6: 5-8.

[4] Gu G, Perdisci R, Zhang J, et al. BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection [C]//USENIX security symposium. 2008, 5(2): 139-154.

[5] Stone-Gross B, Cova M, Cavallaro L, et al. Your botnet is my botnet: analysis of a botnet takeover[C]//Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009: 635-647.

[6] McGrath D K, Gupta M. Behind Phishing: An Examination of Phisher Modi Operandi [J]. LEET, 2008, 8: 4.

[7] Ma J, Saul L K, Savage S, et al. Beyond blacklists: learning to detect malicious web sites from suspicious URLs [C]//Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009: 1245-1254.

[8] Stalmans E, Irwin B. A framework for DNS based detection and mitigation of malware infections on a network [C]//Information Security South Africa (ISSA), 2011. IEEE, 2011: 1-8.

[9] Yadav S, Reddy A K K, Reddy A L, et al. Detecting algorithmically generated malicious domain names [C]//Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. ACM, 2010: 48-61.

[10] Yadav S, Reddy A L N. Winning with DNS failures: Strategies for faster botnet detection [C]//International Conference on Security and Privacy in Communication Systems. Springer, Berlin, Heidelberg, 2011: 446-459.

[11] Antonakakis M, Perdisci R, Nadji Y, et al. From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware [C]//USENIX security symposium. 2012, 12.

[12] Zhou Y, Li Q, Miao Q, et al. DGA-Based Botnet Detection Using DNS Traffic [J]. J. Internet Serv. Inf. Secur., 2013, 3(3/4): 116-123.

[13] Schiavoni S, Maggi F, Cavallaro L, et al. Phoenix: DGA-based botnet tracking and intelligence [C]//International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, Cham, 2014: 192-211.

[14] Tong V, Nguyen G. A method for detecting DGA botnet based on semantic and cluster analysis [C]//Proceedings of the Seventh Symposium on Information and Communication Technology. ACM, 2016: 272-277.

[15] Davuth N, Kim S R. Classification of malicious domain names using support vector machine and bi-gram method [J]. International Journal of Security and Its Applications, 2013, 7(1): 51-58.

[16] Joachims T. Learning to classify text using support vector machines: Methods, theory and algorithms [M]. Norwell: Kluwer Academic Publishers, 2002.

[17] Joachims T. Making large-scale SVM learning practical [R]. Technical report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund, 1998.

[18] HU J, WANG Y, SHI F A N, et al. Malicious Domain Detection Based on Traffic Similarity [J]. DEStech Transactions on Computer Science and Engineering, 2017 (cii).

[19] Banday M T, Qadri J A, Shah N A. Study of Botnets and their threats to Internet Security [J]. Sprouts: Working Papers on Information Systems, 2009, 9(24).

[20] Liang J, Shi Z, Li D, et al. Information entropy, rough entropy and knowledge granulation in incomplete information systems [J]. International Journal of general systems, 2006, 35(6): 641-654.

[21] Shawe-Taylor J, Bartlett P L, Williamson R C, et al. Structural risk minimization over data-dependent hierarchies [J]. IEEE transactions on Information Theory, 1998, 44(5): 1926-1940.

[22] Vapnik V. Principles of risk minimization for learning theory [C]//Advances in neural information processing systems. 1992: 831-838.

[23] Tang Y. Deep learning using linear support vector machines [J]. arXiv preprint arXiv:1306.0239, 2013.

[24] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection [C]//Ijcai. 1995, 14(2): 1137-1145.