Technical research of detection algorithmically generated malicious domain names using machine learning methods

Hieu Ho Duc, Huong Ho Van

***Abstract*— In recent years, many malware use domain generation algorithm for generating a large of domains to maintain their Command and Control (C&C) network infrastructure. In this paper, we present an approach for detecting malicious domain names using machine learning methods. This approach is using Viterbi algorithm and dictionary for constructing feature of domain names. The approach is demonstrated using a range of legitimate domains and a number of malicious algorithmically generated domain names. The numerical results show the efficiency of this method.**

***Tóm tắt*— Trong những năm gần đây, nhiều phần mềm độc hại sử dụng thuật toán sinh tên miền tạo ra lượng lớn các tên miền để duy trì cơ sở hạ tầng mạng ra lệnh và điều khiển (C&C). Trong bài báo này, chúng tôi trình bày một cách tiếp cận để phát hiện tên miền độc hại bằng phương pháp học máy. Cách tiếp cận này sử dụng thuật toán Viterbi và tập từ điển để trích xuất các đặc trưng của tên miền. Cách tiếp cận được thể hiện bằng cách sử dụng một lượng lớn các tên miền hợp pháp và một lượng lớn tên miền độc hại được tạo ra bằng thuật toán sinh tên miền. Các kết quả thực nghiệm đã chỉ ra tính hiệu quả của phương pháp.**

***Keywords*— cyber security; malicious domain; Domain Generation Algorithm; machine learning; deep learning.**

***Từ khóa*— an ninh mạng; tên miền độc hại; thuật toán tạo miền; học máy; học máy sâu.**

This manuscript is received on December 2, 2018. It is commented on December 16, 2018 and is accepted on December 22, 2018 by the first reviewer. It is commented on December 18, 2018 and is accepted on December 27, 2018 by the second reviewer.

I. INTRODUCTION

The Fourth Industrial Revolution has been developing based on artificial intelligence, internet of things and big data. It brings with it a new operational risk for connected and cyber networks. Cyber security is becoming an increasingly prominent problem for businesses and government all over the world. The recent proliferation of cybercrime is perhaps the best evidence and the impact of the biggest incidents can now be felt on a global scale. For instance, there are few better illustrations of the scope of this problem than the recent WannaCry ransomware attack in late 2017 after infecting hundreds of computers, preventing users from accessing the devices without paying a ransom. The damage to the British health service captured headlines in the UK, but the reach of WannaCry was in fact much larger than this - indeed, cyber security firm Kaspersky estimated that more than 45,000 WannaCry attacks were recorded across 74 countries, affecting more than 57,000 people. On October 12, 2016, a massive distributed denial of service (DDoS) attack left much of the internet inaccessible on the U.S. east coast. The attack, which authorities initially feared was the work of a hostile nation-state, was in fact the work of the Mirai botnet. In Vietnam Information Security Day 2017, VNCERT and some security firms presented that the number of botnet poisoned devices used in targeted attacks in Vietnam ranked 4th in the world. It is clear that serious consequences of cyber attacks impacts on economic, political, security.

There are some researches on using machine learning methods for detecting malicious domain names, botnet such as DGA Detection Using Machine Learning Methods [2], Botnet Detection Technology Based on DNS [4], Automatic Analysis of Malware

Behavior using Machine Learning [6], An Evaluation of Machine Learning-based Methods for Detection of Phishing Sites [7], Detection of Algorithmically Generated Malicious Domain [12], Detecting Algorithmically Generated Domains Using Data Visualization and N-Grams Methods [26], Poster: Zero-day Botnet Domain Generation Algorithm (DGA) Detection using Hidden Markov Models (HMMs) [25], Predicting Domain Generation Algorithms with Long Short-Term Memory Networks [27].

Therefore, we want to research and apply machine learning to build effective tool for detecting wide-area botnet code based on domain name resolution data of network operator and early detect DDoS attacks and protect us from cyber security attacks. This work does not seek to compete with the existing tools for detecting DGA but rather it seeks to complement existing works.

In this paper, after Introduction, section II will present cyber security, domain generation algorithm, machine learning methods. After that, machine learning methods section will present about some algorithm and methods for selecting characteristics for building models. Then, section III will present about building domain classification system with machine learning method, which included building dataset, constructing characteristic vectors, system model, evaluation and results. Results consist of two models which use the Viterbi algorithm with logistic regression model and use the Viterbi algorithm with convolutional neural network. Lastly, section IV will present conclusion.

II. BACKGROUND AND RELATED WORK

A. Cyber security

The Domain Name System

The Domain Name System (DNS) is a core component of Internet operation. It ensures the finding of any resource on the internet by just knowing the domain names of URL that is an easy way to remember.

Malware Command and Control

Malware, short for malicious software, is a kind of software that can be installed on a computer without approval from the computer's owner [5]. Cyber-criminals like to maintain control of devices and hosts they compromise for long-term benefits such as financial gain [16]. To achieve this objective, they usually plant some form of a backdoor or create a C&C channel to allow them to re-enter the system at will [17].

Botnet

A botnet is a number of Internet-connected devices, each of which is running one or more bots. Botnets can be used to perform distributed denial-of-service attack (DDoS attack), steal data, send spam, and allows the attacker to access the device and its connection. The owner can control the botnet using command and control (C&C) software.

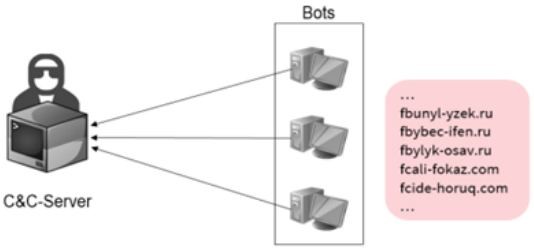

Fig 1. An example of C&C-Server

If the rendezvous points with the command and control servers are hardcoded in the malware the communication can be effectively cut off by blacklisting, which limits the malware's further operation and the extent of their damage. To avoid such static detection mechanisms recent attackers have been taking advantage of various Domain Generation Algorithms (DGA) in choosing and updating the domain names of their command and control servers. DGA embedded in the malware generate a large amount of pseudo-random domain names within a given period, most of which are nonexistent [11].

B. Domain Generation Algorithm

Domain generation algorithms (DGA) are algorithms used to periodically generate a large number of domain names that can be used as rendezvous points with their command and control servers. The large number of domain names generated provides great agility and ensures that even if one or more of the domain names are eventually taken down or blacklisted, the compromised device will ultimately get the IP address of the re-allocated C&C server [18].

```
def generate_domain(year, month, day):
    """Generates a domain name for the given date."""
    domain = ""

    for i in range(16):
        year = ((year ^ 8 * year) >> 11) ^ ((year & 0xFFFFFFF0) << 
        month = ((month ^ 4 * month) >> 25) ^ 16 * (month & 0xFFFFF
        day = ((day ^ (day << 13)) >> 19) ^ ((day & 0xFFFFFFFE) << 
        domain += chr(((year ^ month ^ day) % 25) + 97)

    return domain
```

Fig 2. An example of domain generation algorithm

For example, at its peak the Conficker-C worm, generated almost 50,000 domains per day using DGA [13], [14], [15]. Yet out of the 50,000 domain names generated, Conficker-C only queried roughly 500 of these domains per day.

C. Machine learning methods

Machine learning is the study of algorithms and mathematical models that computer systems use to progressively improve their performance on a specific task.

Supervised learning methods

Supervised learning attempts to discover the relationship between input and its corresponding output. In general, the relationship is represented in a structure, also known as a model that can be used to predict the outputs for some future inputs [19].

Logistic Regression

In statistics, the logistic model is a widely used statistical model that, in its basic form, uses a logistic function to model a binary dependent variable [9].

For the two-class classification problem, we have seen that the posterior probability of class C1 can be written as a logistic sigmoid acting on a linear function of x, for a wide choice of class-conditional distributions p(x|Ck) [9]. The posterior probability of class C1 can be written as a logistic sigmoid acting on a linear function of the feature vector φ so that,

$$p(C1|\varphi) = y(\varphi) = \sigma\,(w^T\varphi)$$

We now use maximum likelihood to determine the parameters of the logistic regression model. To do this, we shall make use of the derivative of the logistic sigmoid function, which can conveniently be expressed in terms of the sigmoid function.

$$d\sigma/da = \sigma(1 - \sigma)$$

Neural Networks

The number of input and outputs units in a neural network is generally determined by the dimensionality of the data set, whereas the number M of hidden units is a free parameter that can be adjusted to give the best predictive performance [9].

In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery [10].

Given a sufficiently large training set, such a network could in principle yield a good solution to this problem and would learn the appropriate invariances by example [9].

Authors in [27] present a DGA classifier that leverages long short-term memory (LSTM) networks for real-time prediction of DGAs without the need for contextual information or manually created features. In addition, the presented technique can accurately perform multiclass classification giving the ability to attribute a DGA generated domain to a specific malware family.

N-grams

In the fields of computational linguistics and probability, an n-gram is a contiguous sequence of n items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus [9].

According to [26], they tested common methods in classification on text strings of domain names has low accuracy. They introduced new features based on N-Grams in the classification methods and our experimental results show that the analysis of N-Gram methods can make a great progress in the accuracy of detection.

Hidden Markov Model

Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved states [9].

According to [25], in this work, they proposed the zero-day DGA detection method using Hidden Markov Models (HMMs). The idea is to compare a DGA-generated domain

name with the HMM that represents the lexical features of the legitimate domain names. By calculating the probability that the given domain name is generated by the HMM, they can decide how likely it is a DGA generated domain name.

III. METHODOLOGY

Building domain classification system with machine learning method.

Construction of the domain classification system.

A. Data Collection

Malicious Domain Set

Collected one million malicious domain names created by various malicious code using DGA such as Zeus, Conficker, Cryptolocker, Goz, etc.

The data was taken from well-known publically available third-party databases to ensure that experimental activity and testing reflected real-life usage and included a previously unknown DGA which are different types of DGA [20].

For example, dsagfdqwf.ru, ofdhiydrpblp.com, madtpojjforoe.biz, etc.

Legitimate Domain Set

Collected one million most popular domain names worldwide and checking for legitimate domains.

The legitimate domains are obtained from Alexa. Alexa has acted as the source of data for a number of other works in this area [21-24].

For example, youtube.com, google.com, amazon.com linkedin.com, etc. Collecting dictionary (100000 words in English).

B. Constructing characteristic vectors

- Using the Viterbi algorithm and using the dictionary for constructing characteristic vectors.

For example, vnexpress.net can be divided into many cases as [vn,express,net], [v,n,express,net], etc and [vn,express,net] is the optimal way of dividing.

Uhbqolxf.org - [u,h,v,q,o,l,x,f,org].

Output is the optimal way of dividing by using graph algorithm and comparing by a sum of squared of lengths of meaningful phrases in dictionary.

- Using the rule to construct a specific vector set.

For example, [vn,express,net] ->[0,1,1,0,1] (the word is defined by the number of the characters, or 1 character equivalent to 0; 2 characters: 1, 3 characters: 1, 4 characters: 0, greater than 5 characters: 1)

uhbqolxf.org => [u, h, b, q, o, l, x, f, org] => [8,0,1,0,0].

After converting domain into characteristic vectors, it is clearly seen that the characteristic vector of this domain names is of large value mainly in low-level weighted scores, while the legit domain names have a large value mainly for distribution of high weighted scores.

C. Building machine learning model

Logistic regression model:

After converting domain into characteristic vectors, we build machine learning model for learning characteristic of two data set.

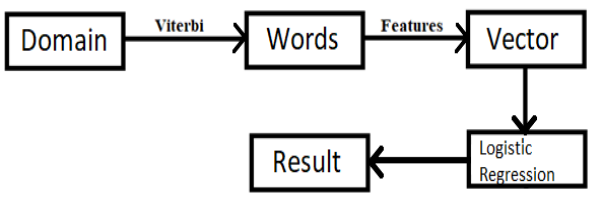

Fig 3. Logistic regression model

Convolutional neural networks model:

After converting domain into characteristic vectors, we use word to vector corresponding to each word and initialize a random vector with the specified space. Then we adjust vector by convolutional neural networks for expressing the relationship between related words.

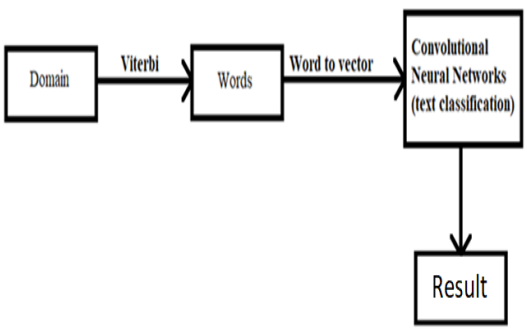

Fig 4. Convolutional neural networks model

D. System model

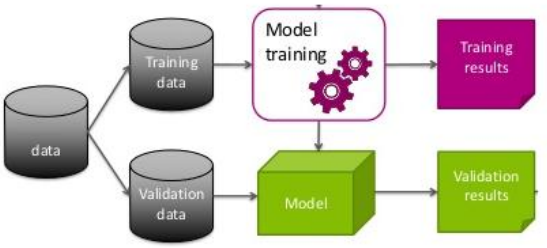

Fig 5. System model

The system model consists of training and testing. The data which include malicious domains and legit domains is divided into training data (70%) and validation data (30%).

Training data is divided into train set and dev set. For every training round, we compare the result with dev set for building the best training model.

After training, we use validation data for evaluating the model.

E. Evaluation

Model is evaluated in the laboratory.

Model is evaluated in the devices which infected malicious.

Model is evaluated in Internet Service Provider network.

F. Results

Model 1: Using the Viterbi algorithm with logistic regression model

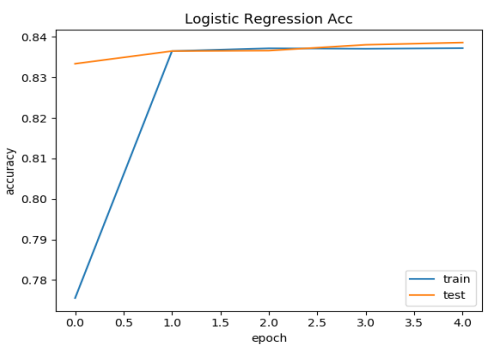

Fig 6. The accuracy of logistic regression model

- The blue line shows the accuracy of logistic regression model on the training set.
- The orange line shows the accuracy of logistic regression model on the test set.
- The x-axis is about the percentage of accuracy.
- The y-axis is about the number of epochs.
- According to the graph, it can be clearly seen that the accuracy of logistic regression model is matching between the training set and testing set after training 0.5 epoch. The model is approximately 83% on the training set and testing set.

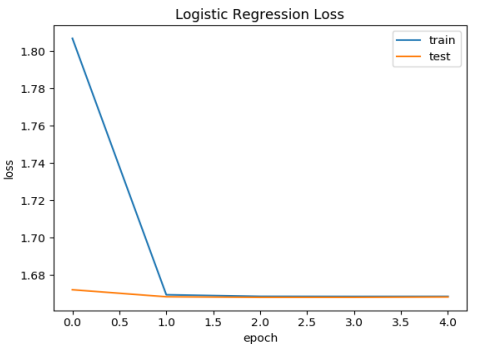

Fig 7. The value of loss function of logistic regression model.

- The blue line shows the value of loss function of Logistic Regression model on the training set.
- The orange line shows the value of loss function of Logistic Regression model on the test set.
- The x-axis is about the percentage of loss.
- The y-axis is about the number of epochs.
- According to the graph, it can be clearly seen that the value of loss function is matching between training set and testing set after training 1 epoch. The value of loss function is approximately 1.68%.

In conclusion, logistic regression model detects malicious domain names with 83% accuracy on the training set and test set*Model 2: Using the Viterbi algorithm with convolutional neural networks*

- The orange line shows the accuracy of model on the training set.
- The blue line shows the accuracy of model on the test set.

Fig 8. The accuracy of convolutional neural networks model

- The x-axis is about the percentage of accuracy.
- The y-axis is about the number of steps for training.
- According to the graph, it can be clearly seen that the model detects malicious domain with 98% accuracy after training 80000 steps.

Fig 9. The loss function of convolutional neural networks model

- The orange line shows the loss function on the training set.
- The blue line shows the accuracy of model on the test set.
- The x-axis is about the percentage of loss.
- The y-axis is about the number of steps for training.
- According to the graph, it can be clearly seen that the value of loss function is approximately 3%.

In conclusion, convolutional neural networks model detects approximately malicious domain names with 98% accuracy after training 80000 steps on the training set and test set.

IV. CONCLUSION

In this paper, we have researched on machine learning with applications in cyber security for detecting malicious domain names. After evaluating and selecting algorithms for optimal system performance and accuracy, we have developed and tested model for detecting malicious domain names with approximately 98% of accuracy in reality at Viettel Cyber Security Lab. Using the Viterbi algorithm combined with the convolutional neural network resulted in better results using the Viterbi algorithm combined with the logistic regression model. The result of this research can apply for detecting wide-area botnet based on domain name resolution data of network operator and early detect DDoS attacks.

In the future, we will apply other machine learning models to assess and analyze with the methodology used to improve performance and accuracy and use large data processing and analysis technologies and optimized machine learning algorithms to contribute to optimal system performance and accuracy.

REFERENCES

[1]. Hà Quang Thụy, Nguyễn Hà Nam, Nguyễn Trí Thành, "Giáo trình khai phá dữ liệu", VNU Publishing, 2013.

[2]. Moran Baruch, "DGA Detection Using Machine Learning Methods", Master Thesis, University of Jyväskylä, 2016.

[3]. Thomas Edgar and David Manz, "Research Methods for Cyber Security", Syngress, 2017.

[4]. Xingguo Li, Junfeng Wang, and Xiao song Zhang, "Botnet Detection Technology Based on DNS", Future Internet 2017, 9, 55.

[5]. Michael Sikorski, Andrew Honig, "Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software", No Starch Press, 2012.

[6]. Konrad Rieck, Philipp Trinius, Carsten Willems, and Thorsten Holz, "Automatic Analysis of Malware Behavior using Machine Learning", 2011.

[7]. Daisuke Miyamoto, Hiroaki Hazeyama, Youki Kadobayashi, "An Evaluation of Machine Learning-based Methods for Detection of Phishing Sites", 2017.

[8]. Jasper Abbink, "Popularity-based Detection of Domain Generation Algorithms, Master Thesis", Delft University of Technology, 2017.

[9]. Christopher M. Bishop, "Pattern Recognition and Machine Learning", Springer Science, 2006.

[10]. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press book, 2016.

[11]. M. Namazifar, Y. Pan, "Research Spotlight: Detecting Algorithmically Generated Domains", Cisco, 2015.

[12]. Enoch Agyepong, William J. Buchanan, Kevin Jones, "Detection of Algorithmically Generated Malicious Domain", Conference: 6th International Conference of Advanced Computer Science & Information Technology, 2018.

[13]. M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou II, S. Abu-Nimeh, W. Lee, and D. Dagon, "From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware". In USENIX security symposium Vol. 12, 2012.

[14]. S. Yadav, A.K.K Reddy, A.L. Reddy, and S. Ranjan, (2010, November). "Detecting Algorithmically generated malicious domain names". In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement pp. 48-61. ACM.

[15]. G. Zhao, K. Xu, L. Xu, and B. Wu, (2015). "Detecting APT Malware Infections Based on Malicious DNS and Traffic A nalysis". IEEE Access, 3, pp. 1132-1142, 2015.

[16]. N. Goodman, "A Survey of Advances in Botnet Technologies". arXiv preprint arXiv:1702.01132, 2017.

[17]. V. Oujezsky, T. Horvath, and V. Skorpil, "Botnet C&C Traffic and Flow Lifespans Using Survival Analysis". International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems, 6(1), pp. 38-44, 201.

[18]. R. Sharifnya, and M. Abadi, DFBotKiller: "Domain-flux botnet detection based on the history of group activities and failures in DNS traffic". Digital Investigation, 12, pp. 15-26, 2015.

[19]. Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques". (2007): 3-24.

[20]. A. Chailytko, and A. Trafimchuk, "DGA clustering and analysis: mastering modern, evolving threats", 2015.

[21]. L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "Finding Malicious Domains Using Passive DNS Analysis", In Ndss, 2011.

[22]. J. Kwon, J. Lee, H. Lee and A Perrig, PsyBoG: "A scalable botnet detection method for large-scale DNS traffic, Computer Networks", 97, pp. 48-73, 2016.

[23]. J. Lee, and H. Lee, "GMAD: Graph-based Malware Activity Detection by DNS traffic analysis", Computer Communications, 49, 33-47, 2014.

[24]. R. Sharifnya, and M. Abadi, "DFBotKiller: Domain-flux botnet detection based on the history of group activities and failures in DNS traffic", Digital Investigation, 12, pp. 15-26, 2015.

[25]. Yu Fu, Lu Yu, Richard Brooks, "Poster: Zero-day Botnet Domain Generation Algorithm (DGA) Detection using Hidden Markov Models (HMMs)", 38th IEEE Symposium on Security and Privacy, 2017.

[26]. Tianyu Wang, Li-Chiou Ch, "Detecting Algorithmically Generated Domains Using Data Visualization and N-Grams Methods", Proceedings of Student-Faculty Research Day, 2017.

[27]. Jonathan Woodbridge, Hyrum S. Anderson, Anjum Ahuja, and Daniel Grant, "Predicting Domain Generation Algorithms with Long Short-Term Memory Networks", arXiv:1611.00791, 2016.

ABOUT THE AUTHORS

Hieu Ho Duc

Workplace: High School for Gifted Students, Hanoi University of Science, Vietnam National University.

Email: hoduchieu01@gmail.com

The education process: He is currently a student at the High School for Gifted Students, Hanoi University of Science, Vietnam National University.

Research today: machine learning, deep learning applied to information security and cyber security.

Dr. Huong Ho Van

Workplace: Vietnam Government Information Security Commission.

Email: huonghv@bis.gov.vn

The education process: He graduated from Academy of Cryptographic Techniques in 1994, Hanoi University of Science and Technology in 1999, received a master's degree in 1999 at Hanoi National University, received a PhD degree at Hanoi National University in 2004.

Research today: cryptography, information security.