Research on Domain Generation Algorithms and Their Detection

Zengfeng Zhou [a], Liyu Zhu [b,] *

Guangdong University of Science and Technology, Dongguan 523000, China;

[a]developer_zzf@outlook.com, [b]zhuliyu6@foxmail.com

* Corresponding Author

Abstract

The paper comprehensively examines the Domain Generation Algorithms (DGA) and their application and detection strategies in the field of cybersecurity. DGA is a mechanism used by malicious software to covertly communicate with its Command and Control (C&C) servers, achieved through generating a large number of pseudo-random domain names. The paper begins by discussing the basic working principles of DGA and classifies its various types, including seed-based DGA, algorithm-based DGA, and hybrid DGA. It then delves into the methods of detecting malicious domains, ranging from rule-based approaches to advanced machine learning and deep learning techniques. Particularly, the paper highlights the potential of Convolutional Neural Networks and Long Short-Term Memory networks in enhancing the efficiency and accuracy of detection. Integrating these analyses, we aim to reveal the current state, challenges, and future trends of DGA detection technology.

Keywords

DGA, Malicious Domains, Machine Learning, Deep Learning.

1. Introduction

In the digital age, cybersecurity has become a focal point of global concern. Particularly, the rapid development of malicious software technologies poses an increasingly serious threat. Among these threats, the use of Domain Generation Algorithms (DGA) by malicious software is notably prominent. DGA is a technique employed by malicious software to dynamically generate a large number of pseudo-random domain names, which serve as communication bridges between the malware and its Command and Control (C&C) servers. These generated domain names enable the malware to receive instructions, upload data, or download malicious code, thus maintaining the continuity and stealth of its network [1].

The characteristic unpredictability and variability of DGA domains render traditional security defenses, such as blacklists and whitelists, ineffective. Consequently, researching and developing effective methods for the detection of DGA domains have become crucial tasks in the field of cybersecurity. This involves not only a deep understanding of the mechanisms behind the generation of malicious domains but also the exploration and evaluation of various detection techniques.

This paper aims to provide a comprehensive overview of DGA and their detection methods. Initially, the paper introduces the working principles of DGA, including the characteristics of different types of DGA such as Seed-based, Algorithm-based, and Hybrid DGA. Subsequently, the paper will explore methods for detecting malicious domains, encompassing rule-based, machine learning, and deep learning approaches. While rule-based methods like blacklists and whitelists are straightforward, they exhibit clear limitations in addressing DGA domains. Therefore, we will focus on detection technologies based on machine learning and deep learning, particularly the application of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks in the detection of DGA domains [2].

By delving into the study of DGA and related detection methods, this paper aims to provide a more comprehensive understanding and powerful tools for the field of cybersecurity, to protect networks from the invasion of malicious domains.

2. Malicious Domain Generation Algorithms

2.1. Working Principle of DGA

DGA are algorithms used to generate pseudo-random and unpredictable domain names, commonly employed by malicious software for communication with their Command and Control (C&C) servers [3]. The working principle of DGA is as follows: The input to a DGA typically includes a seed value, which is associated with the current date to ensure the uniqueness of the domain names generated each time. The seed value plays a crucial role in the DGA as it is the starting point of randomness. Using a Pseudo-Random Number Generator (PRNG) with the seed value as the initial input, the DGA generates a series of random characters. These characters are produced by PRNG through a series of mathematical operations, making them appear random and unpredictable. The complexity of the PRNG and the design of the algorithm are vital for generating domain names with sufficient randomness. The generated random characters are then concatenated to form a complete domain name, typically comprising both a main domain and a subdomain section. The DGA generation algorithm determines which characters will be used for the domain name construction and their order. A key feature is that the seed value can change over time in the DGA algorithm. This means that the domain names generated at different times will be different, as different seed values will lead to the generation of different character sequences. Attackers often change the seed value regularly to ensure that the generated domain names remain unpredictable.

2.2. Classification of DGA Domains

The complexity and diversity of DGA have led to a variety of variants, typically based on seeds, algorithms, or a combination of both in Hybrid DGAs.

2.2.1. Seed-based DGA

The seed is one of the input parameters of the DGA algorithm shared between the attacker and the client malicious software, with different seeds producing different DGA domains. Seeds used in DGA are varied, including dates, trending words from social networks, random numbers, or dictionaries. DGA generates a string of character prefixes based on the seed, which, when appended with a Top-level Domain (TLD), forms the final Algorithmically Generated Domain (AGD). Seeds can be classified as either "time-dependent" or "time-independent," and as either "deterministic" or "non-deterministic". Thus, seed-based DGA domains can be categorized into four types:

TID: time-independent and deterministic.

(2) TDD: time-dependent and deterministic.

(3) TDN: time-dependent and non-deterministic.

(4) TIN: time-independent and non-deterministic.

2.2.2. Algorithm-based DGA

Based on algorithms, DGA domains can be classified into four types [4], as follows:

(1) Arithmetic-based DGA domains, where the algorithm generates a set of values representable in ASCII codes to create DGA domains. This type is most popular.

(2) Hash-based DGA domains, where hexadecimal hash values are used to generate DGA domains. Common hash algorithms include MD5 and SHA256.

(3) Wordlist-based DGA domains, which are generated by combining words selected from proprietary dictionaries, typically starting with a random verb followed by a random noun. These domains reduce the randomness in domain character composition, closely resembling legitimate domains, thereby increasing detection difficulty.

(4) Permutation-based DGA domains, created by randomly generating strings. Different samples generate different strings as domain names. This type of DGA domain is relatively easier to detect as they are often nonsensical strings.

2.2.3. Hybrid DGA

In addition to the above types, there are Hybrid DGAs, which combine features of multiple DGA variants. Hybrid DGAs use multiple input parameters to make the generated domains more diverse and unpredictable. Each variant has its unique generation pattern and characteristics, requiring security researchers to have an in-depth understanding of different DGA variants. A classification diagram of DGA is shown in Fig 1.

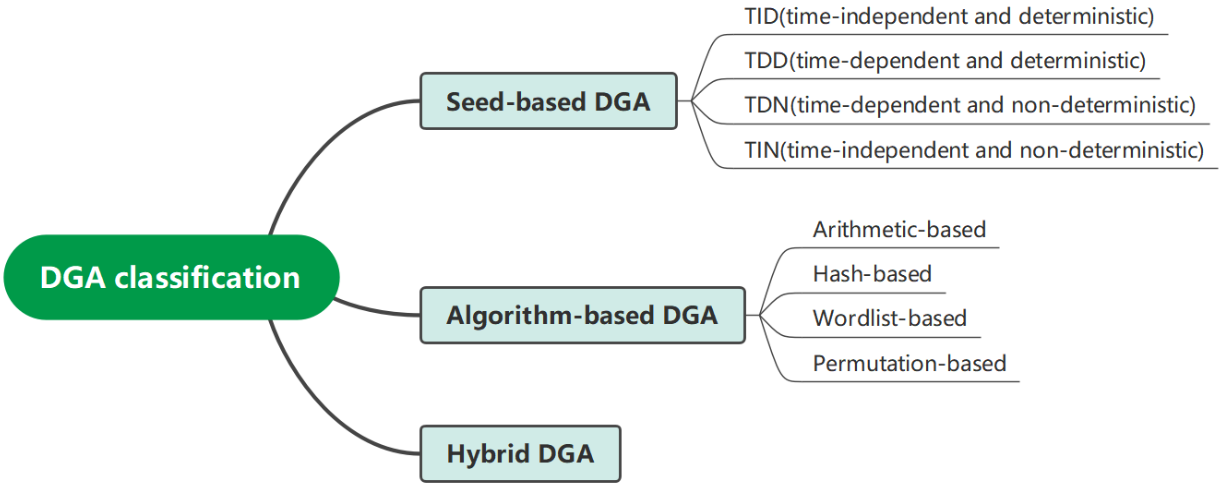
DGA classification
Seed-based DGA
TID(time-independent and deterministic)
TDD(time-dependent and deterministic)
TDN(time-dependent and non-deterministic)
TIN(time-independent and non-deterministic)
Algorithm-based DGA
Arithmetic-based
Hash-based
Wordlist-based
Permutation-based
Hybrid DGA

Fig 1: Classification of DGA Domains

2.3. Application of DGA in Malicious Software

The application of DGA domains in malicious software is extensive and is typically employed for the following purposes:

(1) Control and Communication of Malicious Software: Attackers utilize domain names generated by DGA as addresses for C&C servers, enabling the malicious software to communicate with the attacker's servers. This communication allows the malware to receive commands and upload stolen data.

(2) Infection Spread: DGA can be used to facilitate the spread of malware to other devices or systems. By continuously generating new C&C domain names, the malicious software can propagate to a larger number of victims.

(3) Sustained Threat: The domain names generated by DGA are dynamic and difficult to be captured by traditional domain blacklists. Moreover, DGA aids malicious software in maintaining the continuity of its network. Even if some of the domains are identified and blocked, it does not affect the operation of the entire network. This characteristic allows the malicious software to maintain its stealth and persistency, posing a continuous threat to infected systems.

2.4. Detection Challenges

Detecting domain names generated by DGA presents significant challenges. Firstly, due to the vast number of domains generated at a rapid pace, methods based on blacklists are unable to update and block them in a timely manner. Secondly, the high randomness of DGA domains makes it difficult for traditional pattern-based detection techniques to effectively identify them.

Additionally, as the variants of DGA continue to evolve, the domains they generate become increasingly difficult to distinguish from normal domains. This necessitates continual updates and improvements to the detection algorithms of security systems.

In recent years, machine learning and deep learning technologies have been introduced into DGA domain detection. These technologies enhance the accuracy and efficiency of detection by analyzing the structural features of the domains, their generation patterns, and related network traffic characteristics. Despite these advancements, the continual improvement of DGA strategies by malicious software creators means that detecting DGA remains a significant challenge in the field of cybersecurity.

3. Methods of Detecting Malicious Domains

There has been significant research progress both domestically and internationally regarding DGA and their detection methods. Researchers have analyzed the generation methods of different types of DGA domains to continuously improve detection techniques. Initially, the primary focus was on rule-based detection methods. Currently, the main detection methods are centered around machine learning and deep learning. However, machine learning, heavily reliant on feature extraction and manual analysis, struggles to meet the security needs of real-time detection and faces difficulties in detecting constantly evolving DGA domains. Consequently, current research trends are primarily focused on using deep learning models, particularly CNN and LSTM networks.

3.1. Rule-Based Detection Methods

3.1.1. Blacklist

The blacklist method operates on a predefined list of malicious domains. When a domain matches an entry in the blacklist, the system flags it as malicious. Typically, blacklists are maintained by cybersecurity organizations and are regularly updated to include newly discovered malicious domains. This method is simple and fast in execution, suitable for known malicious domains. Its main drawback, however, is the inability to respond promptly to emerging threats. As blacklists require regular updates, new malicious domains might be active for some time before being identified and added. Shewale et al. pointed out that the blacklist method struggles to respond timely to zero-day attacks since attackers frequently change their techniques and domains. Additionally, this method may lead to a higher false-positive rate, especially when legitimate domains are mistakenly listed in the blacklist.

3.1.2. Whitelist

The whitelist method is based on a list of pre-approved safe domains. This approach can effectively prevent attacks from unknown malicious domains, as only domains on the whitelist are allowed. Cybersecurity organizations create and maintain whitelists based on trustworthiness and safety standards. While the whitelist method can effectively prevent attacks from unknown malicious domains, its main drawback is that it might prevent users from accessing legitimate but not whitelisted websites. Its effectiveness is higher than that of a blacklist, but it lacks flexibility. Soh C Z Y [5] mentioned that the whitelist method could lead to legitimate applications or websites being incorrectly classified as malicious, thus restricting normal user access. Additionally, maintaining a whitelist requires a significant amount of manpower and resources, particularly in a rapidly changing online environment.

3.2. Machine Learning-Based Detection Methods

3.2.1. Feature Engineering

Effective feature engineering can significantly enhance the accuracy of models in identifying malicious domains. In the detection of malicious domains, features such as domain length, character frequency, and usage of dictionary words are utilized for training models. Feature engineering is crucial in machine learning, but its main challenge lies in the need for extensive expertise and experimentation to determine which features are effective. Qihong et al. [6] noted that incorrect or insufficient feature selection could lead to poor model performance. Moreover, with the constant changes in the online environment, feature engineering needs regular updates to adapt to new threat patterns.

3.2.2. Common Machine Learning Algorithms

Common machine learning algorithms used for identifying malicious domains include Support Vector Machines (SVM) [7], Random Forests [8], and others. These algorithms are capable of processing large amounts of data and learning recognition patterns from complex data features. Although machine learning algorithms are effective in the detection of malicious domains, they typically require extensive training data and are highly sensitive to data quality. Hartmuth Ihrig in 《Editorial》 mentioned that inaccurate or biased training data could lead to false positives or negatives in the model. Additionally, these algorithms may require significant computational resources, especially when handling large-scale data.

3.3. Deep Learning-Based Detection Methods

3.3.1. Convolutional Neural Network

CNN [9] is a deep learning algorithm primarily used in fields such as image and video processing, and speech recognition. It extracts features from input data through convolution operations and pooling, and performs tasks like classification or regression using a structure composed of multiple convolutional and fully connected layers. Yu B et al. [10] compared the accuracy of stacked CNN and parallel CNN models in the detection of DGA domains, finding that both CNN models outperform traditional machine learning models such as Random Forests and Multilayer Perceptrons, with the parallel CNN model showing higher accuracy. Luhui Yang et al. [11] further improved the CNN model by adding additional convolutional branches to extract deeper character features from domains, enabling better integration of shallow and deep features, thus enhancing the detection capability for complex domain samples. However, Soh C Z Y [5] mentioned that while CNNs excel in extracting local features, they might not be adept at capturing long-term dependencies, which could be a limitation in the detection of malicious domains.

3.3.2. Long Short-Term Memory Network

The Recurrent Neural Network (RNN) [12] is a type of neural network model used for processing sequential data, but it suffers from issues like vanishing or exploding gradients. As a result, LSTM networks [13] were proposed to better handle long sequence data. The core idea of LSTM is the introduction of a state variable known as the "Memory Cell," capable of storing and conveying information. At each time step, the LSTM calculates the current output and a new state based on the current input and the state from the previous moment. By introducing forget gates, input gates, and output gates, LSTM can better control the flow of information, preventing over-reliance on past or irrelevant information.

Woodbridge J et al. [14] proposed an LSTM-based classifier for DGA domains capable of real-time prediction without the need for contextual information or manual feature creation. Qiao Y et al. [15] introduced an attention mechanism, further enhancing the detection rate for DGA domains based on random characters, though with limitations in detecting word-combination-based DGA domains. Weiqiu Huang et al. [16] combined CNN and LSTM models to improve the detection of dictionary-type DGA domains. These models effectively learned word features, combinations between words, and key character information to address different types of DGA domains. Hybrid methods, like those by Tuan T.A et al. [17], combining CNN and LSTM or adding attention mechanisms, improved the detection rate for various types of DGA domains but still faced challenges in some cases, such as high false-positive rates or deficiencies in multi-class detection. Additionally, researchers have been working to address the multi-class imbalance problem in DGA domain detection. Tran D et al. [18] proposed an algorithm that cleverly combines binary and multi-class models, particularly suitable for detecting malicious domains with a small number in training data.

Overall, these advancements demonstrate the potential of deep learning techniques in enhancing the accuracy of DGA domain detection. These methods hold promise in helping the cybersecurity field better address constantly evolving threats, although they still face some technical and application challenges.

4. Conclusion

As a key component of malicious software, the detection of DGA holds significant importance in the field of cybersecurity. This paper has provided a comprehensive analysis and discussion of the working principles, classification, and current detection techniques of DGA. Regarding the classification of DGA, the paper has detailed the characteristics and differences of seed-based, algorithm-based, and hybrid DGAs. These classifications highlight the diversity and complexity of DGA, underscoring the need for specific detection strategies tailored to different types of DGA.

In terms of methods for detecting DGA domains, this paper has discussed approaches ranging from rule-based methods to advanced techniques based on machine learning and deep learning. While rule-based methods are effective in simple scenarios, they fall short in addressing the complex and variable challenges posed by DGA. In contrast, methods based on machine learning and deep learning, particularly CNN and LSTM, demonstrate stronger adaptability and accuracy. These methods enhance detection efficiency and accuracy by deeply learning the features of DGA domains.

However, in the face of continually evolving DGA techniques, detection methods still need ongoing optimization and updates. Although deep learning technologies show great potential in improving detection performance, they also face challenges in practical application, such as the need for extensive training data and difficulties in dealing with unknown types of DGA. Future research should continue to explore more efficient and accurate detection methods, especially strategies for dealing with unknown and complex types of DGA.

Acknowledgements

This research was supported by the Research Projects of Guangdong University of Science and Technology under Grant No. GKY-2022KYYBK-31.

References

[1] Berge R .The evolving cybersecurity threat[J].Digital Energy Journal, 2016(TN.62).

[2] H. Shahzad. DGA Domain Detection using Deep Learning[C]. International Conference on Cryptography, Security and Privacy (CSP 2021), 2021: 139-143.

[3] Sood A.K., Zeadally S. A Taxonomy of Domain-Generation Algorithms[J]. IEEE Security and Privacy Magazine, 2016, 14(4): 46-53.

[4] Bingrong Chu.Research and Application of improved CNN-LSTM Algorithm in DGA Malicious Domain Name Detection[D]. Hainan Normal University, 2023.

[5] Soh C Z Y. Program analysis and machine learning techniques for mobile security[D].Nanyang Technological University, 2019.

[6] Qihong Shao, Pignataro C M. Self-Adaptive Anomaly Detection With Deep Reinforcement Learning and Topology[J].Technical Disclosure Commons, 2021.

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

[8] Hongkai Wang, et al. . DGA Domain Name Detection Method Based on Random Forest [P] China: CN105577660A, 2016-05-1.

[9] Kim Y. Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv, 2014.

[10] Yu B, Jie P, Hu J, et al. Character Level based Detection of DGA Domain Names[C]. 2018 International Joint Conference on Neural Networks (IJCNN), 2018

[11] Luhui Yang , et al. An improved convolutional neural network malicious domain name detection algorithm [J]. Journal of Xidian University, 2020, 47 (01): 37-43

[12] Grossberg S. Recurrent neural networks[J]. Scholarpedia, 2013, 8(2): 1888.

[13] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[14] Woodbridge J, Anderson H.S., Ahuja A, et al. Predicting Domain Generation Algorithms with Long Short-Term Memory Networks[J], 2016.

[15] Qiao Y, Zhang B, Zhang W, et al. DGA domain name classification method basedon long short-term memory with attention mechanism[J]. Applied Sciences, 2019, 9(20): 4205.

[16] Weiqiu Huang , et al. A Word DGA Domain Name Detection Method Based on APCNN and BiGRU Att [J]. Application Research of Computers, 2021, 39 (5).

[17] Tuan T.A., Long H.V., Taniar D. On detecting and classifying DGA botnets and their families[J]. Computers & Security, 2022, 113: 102549.

[18] Tran D, Mac H, Tong V, et al. A LSTM based framework for handling multiclass imbalance in DGA botnet detection[J]. Neurocomputing, 2018, 275: 2401-2413.