Detection of Malicious Domain Name Based on DNS Data Analysis

Kunsan Zhang1*, Wen Ji2, Nan Li3, Yiting Wang2, Shengyang Liao4

1 State Grid Zhangzhou Power Supply Company, Zhangzhou, Fujian, 363000, China
2 State Grid Fujian Electric Power Co., Ltd, Fuzhou, Fujian, 350000, China
3 State Grid Putian Power Supply Company, Putian, Fujian, 351100, China
4 State Grid Quanzhou Power Supply Company, Quanzhou, Fujian, 362000, China

*Corresponding author’s e-mail: 1406579947@qq.com

Abstract. This paper describes the research background of malicious domain name detection based on DNS data analysis, obtains DNS data through active domain name data or passive domain name data, then obtains DNS data through knowledge-based methods, detection methods based on machine learning and hybrid methods, and finally puts forward research suggestions.

1. Introduction
Various parts of DNS have been subject to extensive attacks, such as the recent attack in DynDNS [1]. The main ways of protection include using better software security technology, increasing the number of copies of key DNS servers and deploying mitigation tools against DDoS.

Attackers will try to destroy the data provided by legitimate DNS servers to control the flow of resources. It has been more than 25 years since DNS hijacking and DNS poison attacks were known. However, it was not until Kaminsky discovered the deadly vulnerability of DNS in 2008 [2] that the security issues in this area really began to attract people’s attention.

Attackers can also illegally obtain advertising revenue, intercept e-mail, or push malicious content to users by re-registering some popular websites that are about to expire, using users’ trust in the websites.

Some attacks use Cybersquatting to pre-register the domain name of the existing company name or person name, or register the domain name very similar to its name, in order to achieve the purpose of illegally profiting from it, or sell it to the injured company or individual at a high price [3]. Typosquatting, this attack is to use the input errors that users may make when entering domain names to register phishing websites, such as imitating Wikipedia www.wikipedia.org to register phishing websites www.wikipedia.org; Bitsquatting, assuming that the memory and CPU cache are flipped due to environment or manufacturing defects, it may lead to visiting phishing websites similar to the domain name of the target website. Soundsquatting, using users’ confusing errors when inputting similar sounds of domain names to register phishing websites, such as www.wikepadia.org; Combosquatting, by adding other keywords to well-known domain names to construct and register new domain names for malicious acts, such as alipay-login.com imitating Alipay login web pages.
1.1. Protect users from attacks using spoofed DNS
Attackers use malicious software and botnets to carry out illegal communication and malicious activities between hosts and C&C servers, including using TXT files to encode control commands into domain name requests [4], etc. Recently, attackers are taking advantage of the UDP-based features of DNS to carry out so-called Reflected Denial-of-Service attack. Real DNS servers are used to send a large number of useless replies to victims.

1.2. Protect users from attacks using real DNS
The main focus of this review is malicious behavior initiated by using real DNS. Earlier, malicious software obtained commands or stole data by hardcoding the IP address of the C&C server into malicious programs. However, this practice was soon abandoned, because once one of the malicious software was caught, all IP would be cracked and the botnet would be blocked directly, so DNS could be used to avoid the direct use of IP. Similarly, in order to avoid IP blacklist, domain names also need to avoid direct use. In order to make the means more flexible and changeable, attackers have created the following two main technologies.

1.2.1. Domain-Flux
This method enables an IP address to correspond to a plurality of different FQDNs, and the changing domain name makes the rendezvous points of communication between attackers and broilers dynamically change, making it difficult for security defense personnel to shut them down. However, this dynamic domain name generation technology is called Domain Generation Algorithm (DGA). The algorithm uses date, time, random number, dictionary and so on as seeds, generates a new random string prefix through a certain function algorithm, and obtains a new Algorithm-Generated Domain (AGD) after adding TLD. Attackers only need to successfully register an AGD to control the botnet, while defense personnel need to find and block all possible AGD in order to completely shut down the network, which is extremely difficult.

1.2.2. IP-Flux
In a normal DNS server, users make DNS queries for the same domain name. In a long period of time, the results returned are basically the same no matter how many queries are made. Fast-IP is to constantly change the IP address corresponding to a domain name. Querying the domain name deployed by Fast-Flux technology in a short time will get different return results. Although the domain names queried are the same, the IP addresses returned are different, resulting in different server hosts accessed, thus protecting the actual host address of the botnet.

Although the malicious attack technologies introduced above are more difficult, fortunately they often leave some clues in DNS data. Therefore, there is the research topic of how to detect malicious behavior through DNS data analysis, which is also the focus of the following discussion.

2. DNS data analysis
There are many ways or means to detect malicious behaviors, including: analyzing network traffic; Check that content of the web page; Check URL; Comprehensive use of the above technologies; Analyze DNS Data.

There are many advantages in malicious detection by analyzing DNS data. DNS data is only a small part of the whole network data, and the amount of data is moderate for analysis, processing and experiment. Malicious behavior will leave traces in DNS data, so the relationship between domain name and malicious behavior can be analyzed by extracting meaningful features. Many extracted features can be further combined with other relevant supplementary information, providing more sufficient space for analysis. Most DNS traffic data is unencrypted, making it possible to acquire and experiment. At the same time, the upcoming attack can be predicted and defended in advance through continuous DNS data analysis[5].
3. Research Process
The research process can be divided into the following five aspects: DNS data collection; Data enrichment; Algorithm design; Ground Truth; Evaluation Methodology. We can refer to the figure 1.

4. Design of algorithm
It can be discussed from three angles:
- Features: What features are used?
- Methods: Based on what technology is the detection?
- Results: What results have been produced?

4.1. Features
The quality of the extracted features plays a vital role in the success of the experimental method. Although some features may have high experimental accuracy, they are easily invalidated by attackers through some minor modifications. Therefore, features need to perform well in both accuracy and robustness. Choose to compare and explain the characteristics from three angles.

4.1.1. Internal vs. Contextual Features
The concepts of internal features and context features are similar to the passive and active features proposed by Perdisci [6].

Internal features can be directly extracted from DNS RRs without any external supplementary information. For example, "Average Time to Live (TTL) Value of Domain Name" and so on. Internal features have simple advantages, which can often reflect a strong difference between malice and benign. It is often used in DGA detection and some graph-based methods.

The advantage of contextual features is that they can provide more other information than internal features that may contain malicious acts. This information comes from the combination of DNS and external information. For example, "how many ASNs does the IPs corresponding to a domain name contain", this feature needs to know the mapping relationship information of IP-AS.

4.1.2. DNS Dataset Dependent vs. Independent Features
The experimental performance of data dependency characteristics is easily affected by the selected data set. Such as "how many IP's are assigned to a domain name" and "how many ASNs [7] are shared by a pair of domain names".
The experimental performance of data independence characteristics is not easily affected by the selected data set. Such as "the number of hits of a domain name on a well-known search engine" and "the n-gram distribution information of a domain name".

4.1.3. Mono vs. Multi Domains Features

Single Domain Name Feature is generally analyzed for a single domain name. For example, "the number of countries with a domain name". If this feature is used in the experiment, experiments such as training can be better carried out on different data sets.

Multi-domain name feature often establishes the relationship information between a pair of domain names, especially in graph-based methods and clustering methods, which require a large amount of data.

4.2. Testing methods

4.2.1. Knowledge-based Approaches

Knowledge-based methods require some professional knowledge, which generally comes from the exploration of the correlation research of malicious domain name behavior. For example, Sato et al.[8] found that malicious domain names belonging to the same malicious software family tend to send request information at the same time, so the malice of unknown domain names is judged by studying the probability of the common occurrence of unknown domain names and known malicious domain names.

The great disadvantage of this method is that experts often do not have enough experience. On the one hand, they cannot directly dig out high-dimensional relationships; On the other hand, malicious attackers often do some behaviors[9]. However, these methods cannot automatically complete the adjustment, resulting in detection failure.

4.2.2. Machine-Learning-Based Approaches

At present, most of the detection methods based on machine learning are still used, which is a method to mine relationships through data and can be classified as follows according to the data types used.

This algorithm requires all data sets to be marked benign or malicious. The advantage of supervised learning is that it can train and learn simply and efficiently, and the trained model can be directly used for detection[10].

However, as mentioned earlier, only a small part of the data is marked, and it is almost impossible and time-consuming to mark all the data correctly, so only as many data with high credibility as possible can be marked. Secondly, supervised learning is prone to over-fitting, which greatly reduces the generalization performance after replacing data sets.

Semi-supervised learning algorithm is proposed to solve the above problems such as insufficient labeling data. It can learn from both tagged and untagged data[11], where untagged data is used to help machine learning algorithms correct relational assumptions obtained from tagged data.

Among them, graph-based inference method and clustering method are currently the hottest methods for malicious detection based on semi-supervision.

Unsupervised methods get rid of the dependence on data tags. For example, we can select good features that may be related to malicious behavior and divide them into two categories through clustering, and then confirm whether each category is benign or malicious domain name respectively.

Although this kind of method can get rid of the dependence on tag data, it is seldom used in practice because its algorithm is difficult to design.

4.2.3 Mixed method

At present, most of the methods actually used are still hybrid methods. This mixing can be realized by mixing and using various machine learning methods, and can also be realized by mixing and using knowledge and machine learning.
5. Results
The results referred to here are not an introduction to the experimental results obtained after using a certain method, but an explanation of the results. As mentioned earlier, there are many types of malicious acts, including sending spam, generating phishing web pages, serving C&C communications, etc. Therefore, according to the experiment, the results can be divided into the following two categories for explanation respectively. Malicious Behavior Specific Approaches and Malicious Behavior Agnostic Approaches. Indefinite malicious behavior detection does not divide the specific types of malicious behaviors, but macroscopically analyzes various behavior connections of domain names. In contrast, specific malicious behavior detection is to mine the characteristics of specific malicious behaviors.

6. Conclusion
First of all, the amount of data to be processed in the actual scene is likely to be larger than that in the experimental study, so it is necessary to know whether the detection algorithm is still applicable. Some methods of adjusting parameters based on big data can be implemented by means of platforms such as Apache Hadoop or Apache Gift. There is also a way to remove some seemingly less important parts of the data, but this may lead to the loss of some malicious domain names. Therefore, a method that can evaluate the complexity and scalability of the algorithm at the same time is needed.

Secondly, the time consumption before the algorithm experiment also needs to be considered. For example, when determining the malicious nature of a domain name, a large amount of observation and analysis of requested information are required, which takes too much time in the early stage.

Then, it is necessary to know that the attacker will adjust his strategy according to the defense to avoid detection, so the algorithm should be updated in time.

Finally, there is a lack of a systematic method that can quantitatively compare and analyze the efficiency of various domain name detection algorithms. In addition, many researchers are unwilling to disclose their data sets and experimental codes, which makes the experiments unable to be compared and reproduced and also brings challenges.

References
[1] Zhauniarovich Y, Khalil I, Yu T, et al. A Survey on Malicious Domains Detection through DNS Data Analysis[J]. Acm Computing Surveys, 2018, 1(1).
[2] Pieter Agten, Wouter Joosen, Frank Piessens, and Nick Nikiforakis. 2015. Seven months’ worth of mistakes: A longitudinal study of typosquatting abuse. In Proceedings of the Network and Distributed System Security Symposium.
[3] Anirban Banerjee, Md Sazzadur Rahman, and Michalis Faloutsos. 2011. SUT: Quantifying and mitigating URL typosquatting. Comput. Netw. 55, 13 (2011), 3001–3014.
[4] M. T. Khan, X. Huo, Z. Li, and C. Kanich. 2015. Every second counts: Quantifying the negative externalities of cybercrime via typosquatting. In Proceedings of the IEEE Symposium on Security and Privacy. 135–150.
[5] Janos Szurdi, Balazs Kocso, Gabor Cseh, Jonathan Spring, Mark Felegyhazi, and Chris Kanich. 2014. The long “taile” of typosquatting domain names. In Proceedings of the USENIX Security Symposium. 191–206.
[6] Wang Y M, Beck D, Wang J, et al. Strider Typo-Patrol: Discovery and Analysis of Systematic Typo-Squatting[J]. SRUTI, 2006, 6: 31-36.
[7] Nick Nikiforakis, Steven Van Acker, Wannes Meert, Lieven Desmet, Frank Piessens, and Wouter Joosen. 2013. Bitsquatting: Exploiting bit-flips for fun, or profit? In Proceedings of the International Conference on World Wide Web. 989–998.
[8] Nick Nikiforakis, Marco Balduzzi, Lieven Desmet, Frank Piessens, and Wouter Joosen. 2014. Soundsquatting: Uncovering the use of homophones in domain squatting. In Proceedings of the International Conference on Information Security. 291–308.
[9] Christian Rossow. 2014. Amplification hell: Revisiting network protocols for DDoS abuse. In Proceedings of the Network and Distributed System Security Symposium.

[10] Seungwon Shin, Zhaoyan Xu, and Guofei Gu. 2012. EFFORT: Efficient and effective bot malware detection. In Proceedings of the IEEE INFOCOM. 2846–2850.

[11] Ting-Fang Yen and Michael K. Reiter. 2008. Traffic aggregation for malware detection. In Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. 207–227.