Domain Algorithmically Generated Botnet Detection and Analysis

Xiaolin Xu[1,2,3,4], Yonglin Zhou[4], and Qingshan Li[5](✉)

[1] Institute of Computing and Technology,
Chinese Academy of Sciences, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Institute of Information Engineering, Chinese Academy of Sciences,
Beijing 100093, China
[4] Computer Emergency Response Team, Beijing 100029, China
{xxl,zyl}@cert.org.cn
[5] Ministry of Education, Key Laboratory of Network and Software Security
Assurance of Peking University, Beijing 100871, China
liqs@pku.edu.cn

Abstract. To detect domains used by botnet and generated by algorithms, a new technique is proposed to analyze the query difference between algorithmically generated domain and legal domain based on a fact that every domain name in the domain group generated by one botnet has similar live time and query style. We look for suspicious domains in DNS traffic, and use change distance to verify these suspicious domains used by botnet. Then we tried to describe botnet change rate and change scope using domain change distance. Through deploying our system at operators' RDNS, experiments were carried to validate the effectiveness of detection method. The experiment result shows that the method can detect algorithmically generated domains used by botnet.

Keywords: Botnet · DNS · Algorithmically generated domains · Domain-flux

1 Introduction

Botnet consists of many compromised hosts, and realizes control of zombie host through the command and control channel [1]. Utilizing Botnet, an attacker can carry out a series of malicious activities [2]. In order to bypass the security system inspection, to improve their survival ability and to prolong live time, DNS is used for organization and control in many Botnets. In recent years a large number of malwares add domain algorithmically generate technique to their command and control module, such as Conficker [3], Kraken [4, 5], Torpig [6], Srizbi and Bobax.

In this paper, we proposed a method to detect DGA-botnet by analyzing and comparing the difference of domain query characteristics between malicious algorithmically generated domain and legitimate domain. Then we calculate the changing speed of related domain sets to describe and demonstrate botnets changes in the view of DNS.

J. Tian et al. (Eds.): SecureComm 2014, Part I, LNICST 152, pp. 530–534, 2015.
DOI: 10.1007/978-3-319-23829-6_38

2 Related Works

Domain algorithmically generating technique become an emerging trend for botnet. In the early stage researchers often use reverse engineering on the botnet executable code analysis.

Brett Stone-Gross et algot the domain generation algorithm after they have a deep reverse analysis of tropig sample [6].

Reverse engineering technique can accurately understand the domain generation algorithm although the entire analysis needs a lot of time 、 resources and the support of a sample library. [7] propose a method to detect malicious domain name in DNS traffic. They found algorithmically generated domain names had obvious difference with legitimate domain names in the distribution of the characters. KL distance, edit distance and Jaccard index were used with machine learning methods to filter algorithmically generated domain. Antonakakis et al. from Damballa used no existing domain traffic to detect randomly generated domain [8]. They believed that in a botnet, each bot would produce consistent DNS traffic. So they used classification and clustering method for data processing.

In previous studies, most of them used classification or clustering algorithm to handle domain traffic and identified generated domain for malicious behavior. Decision trees, Bayesian and K-nearest neighbor were mainly employed. Bayesian Classification in malicious domain filtering is widely used as it is relatively simple, easy to implement and its satisfying classification performance [9]. [10] use Naive Bayes and k-nearest neighbor to classify the training data and concluded that k-nearest neighbor can achieve better classification results. The methods mentioned above rely on known domain data sets or samples, and the detection coverage is infected by training data.

The domain request from bot have a time and space continuity stability, so we can detect domain based on domain query behavior from bot.

3 Dga Detection

3.1 DGA Detection Based on Domain Query Pattern

Compared with normal domain traffic, DNS traffic generated by botnet account for a small proportion of the entire DNS traffic. Therefore, whitelist was used to reduce the raw traffic size. Algorithm generated domain names used by each bot in one botnet has similar query behavior. We cluster domain names by domain prefix and parsed IP. Then we look for this similar live pattern to each group. Since the domain name generation techniques are widely used, so we differentiate the normal use from malicious use relying on data set changes in domain records.

According to the time sequence, with a fixed period of time, the domain flow was divided into several time cycles. For a given time period T, we extract all domain names to one set indicated by $D = \{d_1, d_2, \ldots, d_n\}$, P_i is parsed IP set for d_i. Two domain generated by one algorithm will meet the following two characteristics: $PRE(d_i) = PRE(d_i)$ and $P_i \cap P_j \neq \boldsymbol{\Phi}$.

The automatic generated domain names need to follow the basic domain name conventions which is admitted by domain service providers. So two domains generated by one botnet may have the same top domain. A major objective of generated domain is for the resource location, so the different domain name will point to the same set of parsed IPs.

We can use graph to describe the relationship between domain names. Each node in the graph stands for a domain name, and when two domain names meet the condition $P_i \cap P_j \neq \boldsymbol{\Phi}$, two nodes have an undirected edge. If two domains have higher coincidence of resolved IP set, it indicates that they associate with each other more closely. The distance function between domain names is defined as follows:

$$\varphi(d_i, d_j) = \frac{P_i \cup P_j}{P_i \cap P_j} \tag{1}$$

The smaller the value of $\varphi(d_i, d_j)$ is, the closer the association between domain names is. Based on conditions mentioned above, cluster domains to one group in every time cycle. Finally each group contains at least one member.

Used in one botnet, the live time of every domain name accounted for only a part of the whole life cycle of the botnet. When the life cycle of one domain name is passed, there will be no bot using it although we can still get resolutions about this domain name. As for a domain, the first time that it appeared in system is $t1$, and last time that it was queried is $t2$, then the live span of this domain name observed by our detection system is $\Delta t = t1 - t2$. For a domain group, calculating live span for all members could get domain active set $T = \{\Delta t_1, \Delta t_2, \ldots, \Delta t_n\}$. The use pattern of domain in botnet determines the set T is a single peak data collection, and the mode of set can be seen as single domain use cycle in the botnet. Calculating the proportion of members that equal to mode value, the higher the proportion is, the more suspicious this domain group is. We use this to filter out suspicious domain group. Given a domain group, let the mode of set T be m, its suspicious degree calculation function is defined as follows:

$$Q(D) = \frac{count(m)}{\sum_{1}^{n} count(\Delta t_i)} \geq \beta \tag{2}$$

By setting different threshold β, we can get suspicious domain name set.

3.2 Domain Records Change Analysis

Botnets and normal network have a great difference between the uses of domain names. For a legitimate domain, the number of sub domains is relatively fixed, and in a long time period, the resources that Domain name pointed to are relatively stable, so the access of user to the domain name will not appear larger fluctuation. Compared to botnet generated domains, legitimate domains have less change in the domain mount and resolution data collection.

Let $D = \{d_1, d_2, \ldots, d_n\}$ represent a domain set that contains n domain, and $P = \{ip_1, ip_2, \ldots, ip_n\}$ contains all resolution IPs for D. Given that the domain set botnet used at t1 was D_{t1}. From t1 to t2, the change in domain set was $\overline{D_{t2} \cap D_{t1}}$. $|D|$ is the number of domain set D. The bigger the value of $|D|$ and $|P|$ is, the greater size botnet have. So the botnet change varies from different time using domain data can expressed as:

$$\frac{|\overline{D_{t2} \cap D_{t1}}|}{|D_{t1} \cup D_{t2}|}$$

As the use of domain technique is not limited to domain algorithmically generate, it can also use IP-flux and domain-flux simultaneously. Therefore, in considering the entire botnet domain data changes, domain resolution collection should be counted. In addition, changes in the collection are also associated with the time, the same membership changes take different time reflect different change in speed. In summary, the function describes botnet change from t1 to t2 is:

$$V(t1, t2) = \frac{\frac{|\overline{P_{t2} \cap P_{t1}}|}{|P_{t1} \cup P_{t2}|} + \frac{|\overline{D_{t2} \cap D_{t1}}|}{|D_{t1} \cup D_{t2}|}}{t2 - t1} \tag{3}$$

When greater the value of V, then faster botnet changes with the domain data.

4 Result Analysis

By using function defined above and adjusting the threshold β, we got different domain list. Then external databases like dnsbl and rbls were used for further confirmation. Some domains were appeared one week later in other malware domain lists. When the threshold is set to 0.9 and above, the false positives dropped to about 5 %.

Based on above output and according to the specified time period (default one day) we extract domain resolution and calculate V. For a legitimate domain, due to its relatively stable network business, the V value in each time segment will be close to zero. While the V value of botnet fluctuate around one constant over a period of time.

Figure 1 show the V of one malicious domain me7ns4.com.

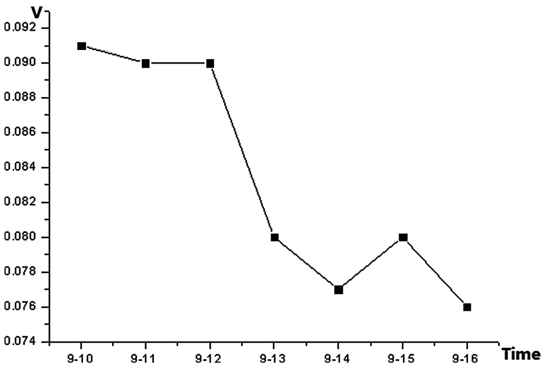

Fig. 1. the V value of me7ns4.com

Adjust V value to filter the domain name, when V is 0.05 or above, the system output is stable.

5 Conclusions

In this paper, we propose a methodology to detect DGA-based botnet, and use distance function to observe filtered domain. Compared to previous works, our approach does not rely on external resources such as known malware domain names and can get some lists earlier than others. But compared to the methods that Antonakakis used, our approach can not accurately group all domains into one set used in one botnet, which needs to be improved in the next step work.

References

1. Abu Rajab, M., Zarfoss, J., Monrose, F., et al.: A multifaceted approach to understanding the botnet phenomenon. In: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, pp. 41–52. ACM (2006)
2. Feily, M., Shahrestani, A., Ramadass, S.: A survey of botnet and botnet detection. In: Third International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2009, pp. 268–273. IEEE (2009)
3. Porras, P., Saïdi, H., Yegneswaran, V.: A foray into Conficker's logic and rendezvous points. In: USENIX Workshop on Large-Scale Exploits and Emergent Threats, pp. 10–11 (2009)
4. Royal, P.: Analysis of the kraken botnet (2008). http://www.damballa.com/downloads/pubs/KrakenWhitepaper.pdf
5. Royal, P.: On the Kraken and Bobax botnets. Whitepaper, Damball, April 2008
6. Stone-Gross, B., Cova, M., Cavallaro, L., et al.: Your botnet is my botnet: analysis of a botnet takeover. In: Proceedings of the 16th ACM Conference on Computer and Communications security, pp. 635–647. ACM (2009)
7. Yadav, S., Reddy, A., Reddy, A., Ranja, S.: Detecting algorithmically generated malicious domain names. In: Proceedings of the 10th Annual Conference on Internet Measurement, pp. 48–61. ACM, Melbourne, Australia (2010)
8. Antonakakis, M., Perdisci, R., Nadji, Y., Vasiloglou, N., Abu-Nimeh, S., Lee, W., Dagon, D.: From throw-away traffic to bots: detecting the rise of DGA-based malware. In: The 21th USENIX Security Symposium, Bellevue, WA, 8–10 August 2012
9. Caglayan, A., Toothaker, M., Drapeau, D., et al.: Real-time detection of fast flux service networks. In: Conference For Homeland Security, 2009. CATCH 2009. Cybersecurity Applications and Technology, pp. 285–292. IEEE (2009)
10. Wu, J., Zhang, L., Liang, J., et al.: A comparative study for fast-flux service networks detection. In: 2010 Sixth International Conference on Networked Computing and Advanced Information Management (NCM), pp. 346–350. IEEE (2010)