A Novel Reputation System to Detect DGA-Based Botnets

Reza Sharifnya and Mahdi Abadi
Faculty of Electrical and Computer Engineering
Tarbiat Modares University
Tehran, Iran
reza.sharifnyay@modares.ac.ir, abadi@modares.ac.ir

***Abstract*—A botnet is a network of compromised hosts (bots) remotely controlled by a so-called bot herder through one or more command and control (C&C) servers. New generation botnets, such as Conficker and Murofet, tend to use a form of domain fluxing for command and control. Each domain fluxing bot generates a list of domain names using a domain name generation algorithm (DGA) and queries each of them until one of them is resolved to a C&C server. Since the bot herder registers only a few of these domain names, the domain fluxing bots generate many failed DNS queries. Even though some efforts have been focused on the detection of DGA-based botnets, but none of them consider the history of suspicious activities. This makes the detection system has a potentially high false alarm rate. In this paper, we propose a novel reputation system to detect DGA-based botnets. Our main goal is to automatically assign a high negative reputation score to each host that is involved in suspicious bot activities. To achieve this goal, we first choose DNS queries with similar characteristics at the end of each time window. We then identify hosts that algorithmically generated a large set of suspicious domain names and add them to a so-called suspicious group activity matrix. We also identify hosts with high numbers of failed DNS queries and add them to a so-called suspicious failure matrix. We finally calculate the negative reputation score of each host in these two matrices and detect hosts with high negative reputation scores as bot-infected. We evaluate our reputation system using DNS queries collected from the campus network. The experimental results show that it can successfully detect DGA-based botnets with a high detection rate and a low false alarm rate while providing real-time monitoring in large-scale networks.**

Keywords-botnet detection; reputation system; suspicious activity; domain name generation algorithm; domain fluxing

I. INTRODUCTION

A botnet is a network of hosts compromised by the same malicious code and remotely controlled by a bot herder through one or more command and control (C&C) servers. The bot herder can use the botnet for sending spam, conducting distributed denial of service (DDoS) attacks, stealing personal information, or other malicious activities [1]. During the last several years, botnets have become one of the serious threats to Internet security. Each botnet needs an addressing mechanism to locate C&C servers. This mechanism allows the bot herder to send commands to and receive stolen data from compromised hosts. New generation botnets use a technique, called domain fluxing, to avoid being shut down by dynamically migrating C&C servers. Domain fluxing botnets, such as Conficker, Kraken, Cycbot, and Murofet, rely on DNS as the service and generate a unique list of domain names based on a predefined algorithm, called domain name generation algorithm (DGA). For example, Conficker.C generates 50,000 domain names every day [2]. Each bot runs the algorithm with a given seed and resolves the generated domain names by sending DNS queries until a domain name is mapped to a C&C server. Therefore, the bots can get the IP addresses of C&C servers even if some domain names of them are blocked. Indeed, DGA-based botnets combine the facility of centralized C&C servers with the power of P2P structures to make their C&C communications more resistant to botnet detection systems or other security measures [3].

Each DGA-based botnet generates the list of domain names in a different way. For example, Conficker-C generates domain names by using the current date and time at UTC as the seed. Kraken generates specific English-language alike words and combines each of them with a randomly chosen suffix, such as -able, -dom, -ment, -ship, or -ly [4]. Since the bot herder registers only a small number of domain names in the list, his bots daily generate a large number of unsuccessful DNS resolutions for non-existent domain names. To predict future domain names, a security vendor has to reverse engineer the DGA algorithm, which is a time-consuming process and during this time the bot herder may command his bots to change the algorithm.

DGA-based botnets can be recognized by the following characteristics [3], [4]: (a) the bots generate a large number of similar DNS queries, (b) the domain names in the DNS queries are generated algorithmically and their alphanumeric distribution is significantly different from human-generated ones, and (c) many of the DNS queries are failed as many of the algorithmically generated domain names may not be registered.

Reputation systems represent a significant trend in supplementary decision making services. The basic idea is to calculate a reputation score for each object within a community or domain, based on a set of opinions held about it. The systems have been extensively used in many applications, such as P2P networks [5], multi-agent systems [6], mobile ad-hoc networks [7], and so on, but few studies to date have applied them for botnet detection.

978-1-4799-2093-8/13/$31.00 ©2013 IEEE

An inherent characteristic of each botnet is group activity [8], [9]. We use this characteristic of botnets in conjunction with the above characteristics to detect DGA-based botnets. To this end, we propose a novel online reputation system that considers the history of both suspicious group activities and failures in DNS traffic to automatically assign a high negative reputation score to each bot-infected host. The history of suspicious activities helps the system to effectively reduce false alarms.

We use DNS traffic for three reasons. First, in DGA-based botnets, each bot daily generates a large number of domain names and resolves them by using DNS queries to obtain the IP address of a C&C server. Because these domain names are algorithmically generated, the probability of detecting bot-infected hosts is increased. Second, DNS traffic is a small percentage of the network traffic. Thus, monitoring DNS traffic has less overhead than monitoring the whole network traffic. Third, DGA-based botnets generate many failed DNS queries in early stages of their life-cycles. Therefore, we can quickly detect bot-infected hosts before performing any malicious activity.

The remainder of this paper is organized as follows. In Section II, we introduce the definitions used in this paper and briefly review related work in Section III. In Section IV, we present our reputation system to detect DGA-based botnets and evaluate it in Section V. Finally, we give some conclusions in Section VI.

II. BASIC DEFINITIONS

As previously mentioned, DGA-based botnets use a technique, called domain fluxing, to protect their C&C infrastructures from takedowns [2]. This technique allows them to bypass the domain blacklists. Each bot algorithmically generates a large number of domain names and queries each of them until one of them is resolved. In the following, we introduce some basic terminology used throughout this paper.

Definition 1 (Domain name). A domain name is an easy-to-remember identification string for a particular web site, application, or service on the Internet. Actually, it is an alias for an IP address.

Domain names have two or more parts, separated by dots (.). For example, the domain name *cs.stanford.edu* consists of three parts. The rightmost part of a domain name is known as the top level domain (TLD). For example, *com*, *edu*, *net*, and *org* are some of the most commonly used top level domains [10]. Second level domain (SLD) is the second part from the right. The third part from the right represents the third level domain (3LD). For example, TLD, SLD, and 3LD in *cs.stanford.edu* are *edu*, *stanford*, and *cs*, respectively. As well as, we call all parts to the left of an SLD as the subdomain of that SLD.

Definition 2 (Domain flux). A technique used by new generation botnets that refers to constantly changing and allocating multiple domain names to a single IP address to foil takedown attempts.

A domain fluxing bot dynamically generates a unique list of domain names based on a predefined algorithm, called domain name generation algorithm (DGA).

Definition 3 (Domain name generation algorithm). An algorithm that automatically generates a list of domain names with an initial seed. Each domain name in this list is resolved by a DNS query until there is no domain name left or a domain name is resolved to a C&C server [11].

In our reputation system, we build two separate matrices, called suspicious group activity matrix and suspicious failure matrix to calculate the negative reputation scores of suspicious hosts.

Definition 4 (n-gram). An n-gram is a contiguous sequence of n alphanumeric characters in a domain name. An n-gram of size 1 is referred to as a "unigram" and size 2 as a "bigram" [12].

Definition 5 (Group activity). A number of hosts participate in a group activity in DNS traffic, if they query domain names that are eventually mapped to the same IP address or have the same TLD and SLD.

Definition 6 (Suspicious group activity). A group activity is called suspicious if queried domain names in this group activity are algorithmically generated.

Definition 7 (Suspicious group activity matrix). The suspicious group activity matrix is a binary matrix used to store time windows in which each host participates in at least one suspicious group activity. This matrix is denoted by $G_{n\times m}$, where n is the total number of hosts in the monitored network and m is the number of time windows. An element $g_{ik} \in G$ is equal to 1 iff the host h_i participates in at least one suspicious group activity within the time window k.

Definition 8 (Suspicious failure). When the domain name in a DNS query is unable to be resolved to an IP address, a failure occurs. A suspicious failure arises when the number of failures generated by a host passes a certain predefined threshold.

Definition 9 (Suspicious failure matrix). The suspicious failure matrix is a binary matrix used to store time windows in which each host generates a suspicious failure. This matrix is denoted by $F_{n\times m}$, where n is the total number of hosts in the monitored network and m is the number of time windows. An element $f_{ik} \in F$ is equal to 1 iff the host h_i generates a suspicious failure within the time window k.

Definition 10 (Suspicious activity). A host has a suspicious activity iff it participates in a suspicious group activity or it has a suspicious failure.

Two hosts are said to be participant if they have suspicious activities in the same time window.

Definition 11 (Negative reputation score). A score between 0 and 1, which is given to a host to indicate the amount of its suspicious activities over different time windows.

The negative reputation score is calculated for each host based on the history of its suspicious activities and similarity of its participants in previous time windows. A host is reported as bot-infected if its negative reputation score exceeds a predefined threshold.

978-1-4799-2093-8/13/$31.00 ©2013 IEEE

III. Related Work

DNS traffic analysis is a promising source to detect malware using DNS queries for conducting malicious activities, especially DGA-based botnets. In this section, we briefly review some DNS-based botnet detection techniques.

Stalmans and Irwin [13] proposed a technique to detect fast fluxing botnets using DNS queries. The technique uses features derived from DNS query responses to build a Naïve Bayesian classifier for classifying a domain name as malicious or legitimate. However, the experimental results show that it can detect malicious domain names with a detection rate of 82% and a false alarm rate of 8.3%, which is not acceptable in real network. Yadav and Reddy [14] presented a technique in which the temporal correlation of both successful and failed DNS queries along with the entropy of domain names belonging to such queries are used to speed up the detection of DGA-based botnets. Yadav *et al*. [4] presented a technique to detect DGA-based botnets by looking at the distribution of unigrams and bigrams in all domain names that are mapped to the same IP address or have the same TLD and SLD. Choi and Lee [8] introduced BotGAD, an online unsupervised botnet detection technique. They define a group activity as a main characteristic of botnets and suggest 13 DNS-based features to capture botnet group activities. BotGAD employs the X-means clustering algorithm to detect correlated domain names. However, it is vulnerable to time-based evasion techniques [15]. All aforementioned works do not consider the history of suspicious activities for each host. This makes them to potentially have a high false alarm rate.

IV. DNS-Based Reputation System

In this section, we propose a DNS-based reputation system for DGA-based botnet detection. Our goal is to assign a high negative reputation score to each bot-infected host that algorithmically generates a large number of domain names and automatically queries each of them to obtain the IP addresses of C&C servers.

Our DNS-based reputation system consists of four main components (as shown in Fig. 1): (1) whitelist filtering, (2) suspicious group activity detector, (3) suspicious failure detector, and (4) negative reputation calculator. It can be deployed at the edge of a monitored network to capture and analyze DNS traffic. In the following, we introduce each of its components in detail.

A. Whitelist Filtering

The whitelist filtering maintains a list of trusted domain names (*e.g.*, *google.com*) and filter out DNS queries based on them to reduce calculation time and false alarms. To make the list, we use the Alexa top 100 sites [16]. Our reason is that the most popular 100 websites on the Internet would not probably be bot-infected.

B. Suspicious Group Activity Detector

As previously mentioned, an inherent characteristic of each botnet is group activity. We can identify two specific group activities in DNS traffic: different hosts can generate a set of domain names (1) resolved to the same IP address or (2) having the same SLD and TLD. A bot herder may assign several domain names to the IP address of a C&C server. These domain names are often algorithmically generated. For example, Fig. 2 shows some domain names generated by Conficker.C. As well as, bot herders can see several advantages in generating subdomains of SLDs. A bot herder may purchase the domain name *malicious.com* from a registrar and then freely create the subdomains *dga1.malicious.com*, *dga2.malicious.com*, and so on to avoid increased costs. Also, even if traffic to a subdomain is blocked, traffic to other subdomains within the same SLD is not blocked. For example, Fig. 3 shows some domain names generated by Cycbot.

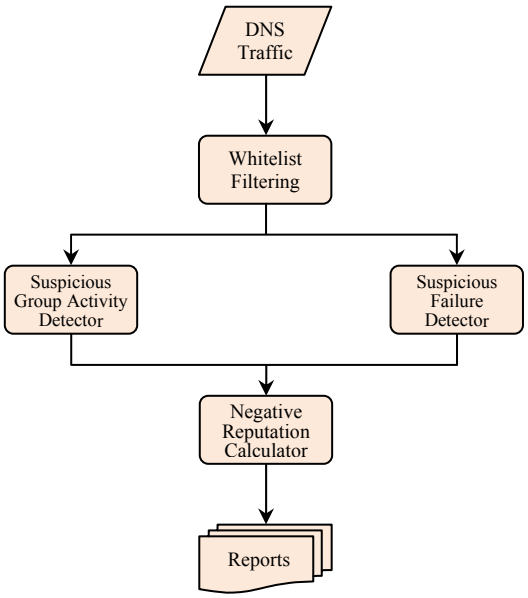


Figure 1. The architecture of our DNS-based reputation system

nsgwaptfpb.info
yntuduawff.biz
mwolcungru.org

Figure 2. Domain names generated by Conficker.C

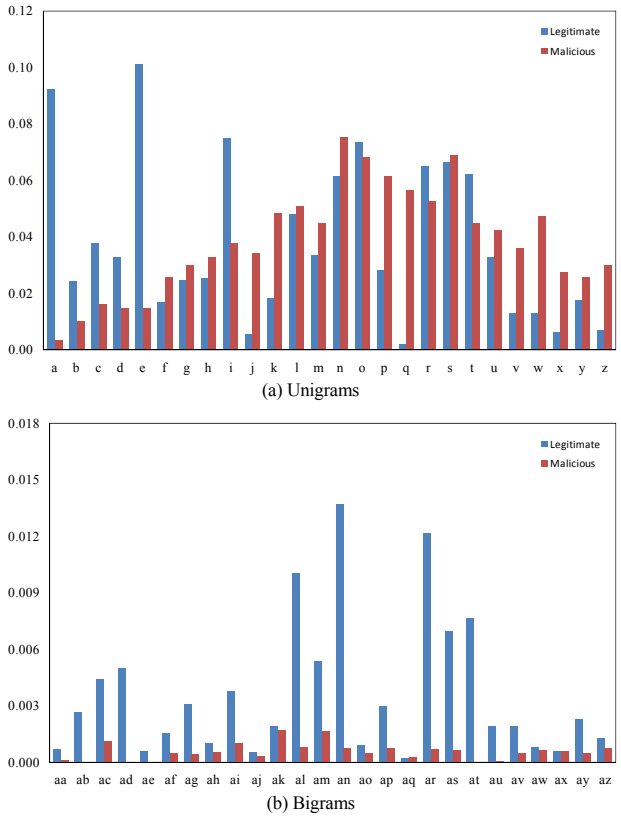
sbro1473vh5d.datamediaarchive.com
5cjjd7m7ujsid.datamediaarchive.com
v61gx269hg5.datamediaarchive.com

Figure 3. Domain names with the same SLD generated by Cycbot

Domain names or subdomains generated by DGA-based botnets have different distributions of n-grams compared to those of legitimate ones [4]. Hence, we mark a domain name as algorithmically generated if the distribution of n-grams of this domain name deviates from the normal distribution. Fig. 4 shows the distribution of unigrams and bigrams in legitimate and malicious domain names. We obtained the legitimate domain names from the Alexa top 1,000,000 sites [16] and the malicious domain names from Murofet.

At the end of each time window, we identify suspicious group activities in DNS traffic and add hosts participating in these activities to the suspicious group activity matrix. We mark a group activity as suspicious if domain names in this group activity are algorithmically generated.

978-1-4799-2093-8/13/$31.00 ©2013 IEEE



(a) Unigrams

(b) Bigrams

Figure 4. Frequencies of some unigrams and bigrams in legitimate and malicious domain names

We use two suspicious measures to identify algorithmically generated domain names:

1) Kullback-Leibler divergence: The Kullback-Leibler (K-L) divergence is a non-symmetric measure of distance between two probability distributions. For discrete probability distributions P and Q, the K-L divergence of Q from P, denoted by $D_{KL}(P \parallel Q)$, is defined as

$$D_{KL}(P \parallel Q) = \sum_{i=1}^{l} P(i) \cdot \log \frac{P(i)}{Q(i)} \ , \qquad (1)$$

where l is the number of possible values for a discrete random variable. We refer to P as the test distribution and Q as the base distribution. Since the K-L divergence is inherently asymmetric, we use a symmetric form [4] of it, which is calculated as

$$D_{SKL}(P, Q) = \frac{1}{2}(D_{KL}(P \parallel Q) + D_{KL}(Q \parallel P)) \ . \qquad (2)$$

Given a group of test domain names for which we want to determine whether they are algorithmically generated or not, we first calculate the distribution of n-grams in this group to obtain the test distribution. We then calculate the symmetric K-L divergence with a base distribution obtained from a group of legitimate domain names. If the measure is greater than a threshold ϑ, we classify the test domain names as algorithmically generated.

2) Spearman's rank correlation coefficient: In statistics, the Spearman's rank correlation coefficient (SRCC), denoted by ρ, is a non-parametric measure of statistical dependence between two ranked variables [17]. We use this measure to calculate the correlation between two different frequency rankings of a list of n-grams. To obtain the frequency ranking of n-grams in a group of domain names, we assign a ranked value to each n-gram based on the number of its occurrences in the group. Suppose we are given the frequency ranking of n-grams in a group of legitimate domain names. We refer to this ranking as the base ranking. To determine whether a group of test domain names are algorithmically generated or not, we first obtain the so-called test frequency ranking of n-grams in this group and then calculate the correlation between the base and test frequency rankings as

$$\rho = \left| 1 - \frac{6}{l(l^2 - 1)} \sum_{i=1}^{l} d_i^2 \right| \ , \qquad (3)$$

where l is the number of unique n-grams and d_i is the difference between the ranked value of n-gram i in the base and test frequency rankings. The value of ρ ranges from 0 to 1. A value close to 0 implies little or no correlation and a value close to 1 implies high correlation. If the correlation is less than a threshold σ, we classify the test domain names as algorithmically generated.

C. Suspicious Failure Detector

When the domain name in a DNS query is unable to be resolved to an IP address, a failure occurs. As previously mentioned, DGA-based botnets generate many failures in early stages of their life-cycles. The high number of failures in DNS traffic can be considered as a suspicious measure to identify bot-infected hosts. Hence, at the end of each time window, we identify hosts generating suspicious failures and add them to the suspicious failure matrix.

D. Negative Reputation Calculator

After updating the suspicious group activity and suspicious failure matrices, we update the negative reputation scores of hosts having suspicious activities in the current time window. The negative reputation score of a host is calculated based on the history of its suspicious activities and similarity of its participants in previous time windows. The host is reported as bot-infected if its negative reputation score is high.

1) History of suspicious activities: In a monitored network, few observations of suspicious activities are not enough to make a correct judgment about the negative reputation of a host and we need to observe a significant number of suspicious activities before we can say the host is reputable to be bot-infected. As a result, we assign high negative reputation scores to the hosts that have a long history of suspicious activities. Generally, we can identify two different histories of suspicious activities in DNS traffic: the history of suspicious group activities and the history of suspicious failures.

Suppose G is the suspicious group activity matrix. We define $\mathcal{I}_{\mathcal{G}}(h_i, G, t)$ to be the history of suspicious group activities for a host h_i from the time window $t - m$ to t:

$$\mathcal{I}_{\mathcal{G}}(h_i, G, t) = \begin{cases} \sin\left(\frac{\pi}{2\epsilon} \cdot \beta(h_i, G, t)\right) & \beta(h_i, G, t) \in [0, \epsilon) \\ 1 & \text{otherwise} \end{cases} , \qquad (4)$$

where $\epsilon < m$ is a user-specified parameter and $\beta(h_i, G, t)$ is the number of time windows in which we have identified a suspicious group activity for the host h_i:

$$\beta(h_i, G, t) = \sum_{\tau=t}^{t-m+1} g_{i\tau} \ , \ g_{i\tau} \in G \ . \tag{5}$$

As well as suppose F is the suspicious failure matrix. We define $\mathcal{I}_{\mathcal{F}}(h_i, F, t)$ to be the history of suspicious failures for a host h_i from the time window $t - m$ to t:

$$\mathcal{I}_{\mathcal{F}}(h_i, F, t) = \begin{cases} \sin(\frac{\pi}{2\varepsilon} \cdot \gamma(h_i, F, t)) & \gamma(h_i, F, t) \in [0, \varepsilon) \ , \\ 1 & \text{otherwise} \ , \end{cases} \tag{6}$$

where $\varepsilon < m$ is a user-specified parameter and $\gamma(h_i, F, t)$ is the number of time windows in which we have identified a suspicious failure for the host h_i:

$$\gamma(h_i, F, t) = \sum_{\tau=t}^{t-m+1} f_{i\tau} \ , \ f_{i\tau} \in F \ . \tag{7}$$

2) Similarity of participants: Since bots of the same botnet run the same malicious code, we expect a bot-infected host has the same set of participants in different time windows. Hence, we assign high negative reputation scores to the hosts whose diversity of their participants in different time windows is very low.

We use the Jaccard similarity coefficient [18] to measure similarity between the participants of a host h_i in the current and previous time windows:

$$\mathcal{J}_s(h_i, G \vee F, t) = \frac{1}{m-1} \sum_{\tau=t-1}^{t-m+1} \frac{|\delta(h_i, G \vee F, t) \cap \delta(h_i, G \vee F, \tau)|}{|\delta(h_i, G \vee F, t) \cup \delta(h_i, G \vee F, \tau)|} \ , \tag{8}$$

where $\delta(h_i, G \vee F, t)$ and $\delta(h_i, G \vee F, \tau)$ are the sets of participants of h_i in the current and previous time windows t and τ, respectively. The value of $\mathcal{J}_s(h_i, G \vee F, t)$ ranges from 0 to 1. A value close to 0 implies no similarity and a value close to 1 implies high similarity between participants in different time windows.

3) Updating negative reputation score: After calculating the history of suspicious activities and similarity of participants for the host h_i, we update its negative reputation score as

$$\mathcal{R}(h_i, G, F, t) = w_1 \cdot \mathcal{I}_{\mathcal{g}}(h_i, G, t) + w_2 \cdot \mathcal{I}_{\mathcal{F}}(h_i, F, t) + w_3 \cdot \mathcal{J}_s(h_i, G \vee F, t) \ , \tag{9}$$

where $\sum_{i=1}^{3} w_i = 1$. For simplicity, we set the same value to w_1, w_2, and w_3. The host h_i is reported as bot-infected if $\mathcal{R}(h_i, G, F, t)$ is greater than a predefined threshold ξ.

V. Experimental Results

In this section, we evaluate the performance of our reputation system on real-world DNS traffic for different settings of parameters.

To generate malicious DNS traffic, we obtained the latest samples of the Kraken, Cycbot, and Murofet botnets from the Offensive Computing website [19]. We ran each sample on 10 virtual machines connected directly to the Internet and collected 112,528 DNS queries from Kraken, 2,120 from Cycbot, and 19,678 from Murofet. We also collected 2,839,500 benign DNS queries from our campus network.

Recall that our reputation system consists of four main components: whitelist filtering, suspicious group activity detector, suspicious failure detector, and negative reputation calculator. We conducted experiments to evaluate the effect of these components in our reputation system. In all experiments, we set the parameters m to 5, ϵ to 3, and ε to 3. We also set the threshold ξ to 0.6 and the length of a time window Δt to 20 minutes.

We present the results for all the four suspicious measures described earlier, namely, the K-L divergence with unigram distribution (K-L Unigram), the K-L divergence with bigram distribution (K-L Bigram), the SRCC with unigram frequency ranking (SRCC Unigram), and the SRCC with bigram frequency ranking (SRCC Bigram). To determine the values of the thresholds ϑ and σ for these measures, we created a list of 1000 most popular websites which randomly selected from the Alexa top 1,000,000 sites [16].

Fig. 5 and Fig. 6 show the false alarm and detection rates of our reputation system in different time windows for Kraken, Cycbot, and Murofet, respectively. As we can see, for all botnets, the detection rate gradually increases until it reaches 100%. This is because the negative reputation scores of bot-infected hosts increase as they perform suspicious activities. However, if some bot-infected hosts do not perform suspicious activities in consecutive time windows, the detection rate slightly decreases. Moreover, K-L Bigram makes a better trade-off between the false alarm and detection rates.

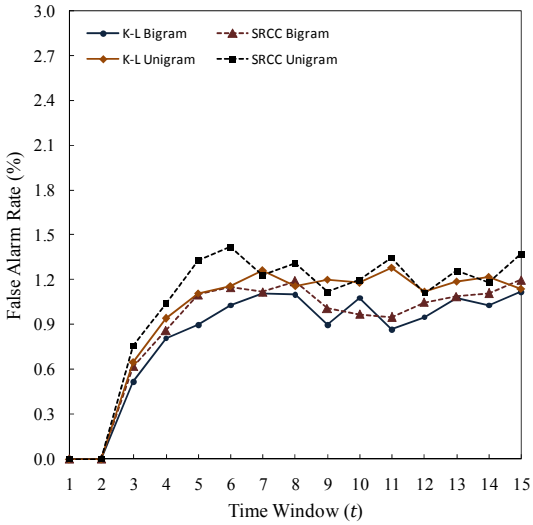
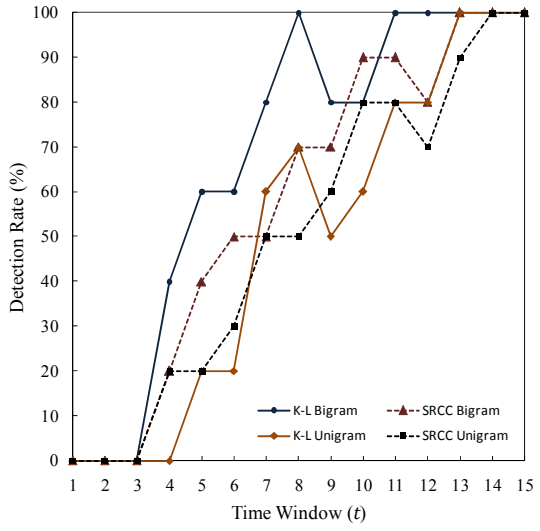
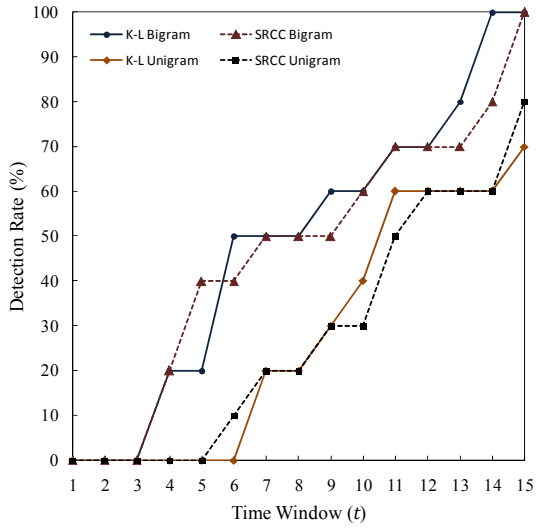


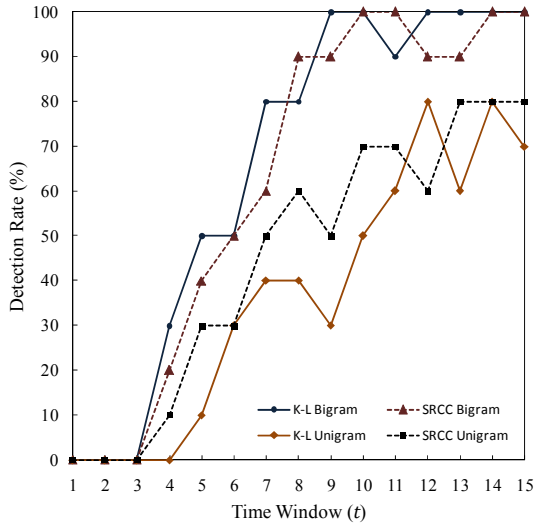
Figure 5. The false alarm rate of our reputation system in different time windows



(a) Kraken



(b) Cycbot



(c) Murofet

Figure 6. The detection rate of our reputation system in different time windows for three botnets

It is not easy to make a fair comparison among various DGA-based botnet detection techniques due to differences between testbed networks, volume of DNS traffic, bot samples used in experiments, and lack of common datasets. Thus, instead of doing a performance comparison between our reputation system and other DGA-based botnet detection techniques, we compare them in terms of some significant characteristics. In Table I, we give a brief comparison between our reputation system and other state-of-the-art DGA-based botnet detection techniques, previously discussed in the related work section. Our reputation system considers the history of both suspicious group activities and failures in DNS traffic to automatically assign a high negative reputation score to each bot-infected host. The history of suspicious activities helps the system to effectively reduce false alarms.

TABLE I. COMPARISON OF OUR REPUTATION SYSTEM WITH OTHER DGA-BASED BOTNET DETECTION TECHNIQUES

Botnet Detection Method	Suspicious Group Activities	Failed DNS Queries	History of Suspicious Activities	Low False Alarm Rate
Stalmans and Irwin [13]	×	×	×	×
Yadav and Reddy [14]	×	✓	×	✓
Choi and Lee [8]	✓	×	×	×
Yadav *et al.* [4]	✓	×	×	×
Our Reputation System	✓	✓	✓	✓

VI. CONCLUSION

In recent years, botnets have become one of the serious threats to Internet security. Thus, it is necessary to provide techniques for botnet detection. New generation botnets tend to use a technique called domain fluxing for locating C&C servers. Each domain fluxing bot generates a large set of domain names based on a predefined algorithm, called domain name generation algorithm (DGA), and queries each of them until one of them is mapped to a C&C server. In this paper, we have proposed an online reputation system to detect DGA-based botnets. Our main goal is to automatically assign a high negative reputation score to each host that is involved in suspicious bot activities. For this purpose, we use statistical measures, such as the Kullback-Leibler (K-L) divergence and the Spearman's rank correlation coefficient (SRCC), to identify hosts that algorithmically generate a large set of suspicious domain names and add them to a so-called suspicious group activity matrix. We also identify hosts with high numbers of failed DNS queries and add them to a so-called suspicious failure matrix. We finally calculate the negative reputation scores of hosts having suspicious activities. A host is reported as bot-infected if its negative reputation score is greater than a predefined threshold.

We performed several experiments using a dataset of benign and malicious DNS queries to evaluate the performance of our reputation system for the four suspicious measures: the K-L divergence with unigram distribution (K-L Unigram), the K-L divergence with bigram distribution (K-L Bigram), the SRCC with unigram frequency ranking (SRCC Unigram), and the SRCC with bigram frequency ranking (SRCC Bigram). The experimental results demonstrated that our reputation system with K-L Bigram makes a better trade-off between the false alarm and detection rates.

978-1-4799-2093-8/13/$31.00 ©2013 IEEE

REFERENCES

[1] J. Goebel and T. Holz, "Rishi: Identify bot contaminated hosts by IRC nickname evaluation," in Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets (HotBots), Cambridge, MA, USA, April 2007.

[2] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "Exposure: Finding malicious domains using passive DNS analysis," in Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, February 2011.

[3] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throw-away traffic to bots: Detecting the rise of DGA-based malware," in Proceedings of the 21st USENIX Security Symposium, Bellevue, WA, USA, pp. 24–40, August 2012.

[4] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, "Detecting algorithmically generated domain-flux attacks with DNS traffic analysis," IEEE/ACM Transactions on Networking, vol. 20, no. 5, pp. 1663–1677, October 2012.

[5] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks," in Proceedings of the 12th International Conference on World Wide Web, Budapest, Hungary, pp. 640–651, May 2003.

[6] S. Jordi and C. Sierra, "REGRET: Reputation in gregarious societies," in Proceedings of the 5th International Conference on Autonomous Agents, Montreal, Canada, pp. 194–195, May 2001.

[7] M. Ibrohimovna and S. Heemstra, "Reputation-based systems within computer networks," in Proceedings of the 5th International Conference on 5th Internet and Web Applications and Services (ICIW), Barcelona, Spain, vol 9, no 15, pp. 96–101, May 2010.

[8] H. Choi and H. Lee, "Identifying botnets by capturing group activities in DNS traffic," Computer Networks, vol. 56, no. 1, pp. 20–33, January 2012.

[9] M. Yahyazadeh and M. Abadi, "BotOnus: An online unsupervised method for botnet detection," The ISC International Journal of Information Security (ISeCure), vol. 4, no 1, pp. 51–62, January 2012.

[10] J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, Upper Saddle River, NJ: Pearson Education, March 2012.

[11] T. Barabosch, A. Wichmann, F. Leder, and E. Gerhards-Padilla, "Automatic extraction of domain name generation algorithms from current malware," in Proceedings of the NATO Symposium IST-111 on Information Assurance and Cyber Defense, Koblenz, Germany, September 2012.

[12] S. Banerjee and T. Pedersen, "The design, implementation, and use of the n-gram statistics package," in Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing, Mexico City, Mexico, pp. 370–381, February 2003.

[13] E. Stalmans and B. Irwin, "A framework for DNS based detection and mitigation of malware infections on a network," in Proceedings of the Information Security South Africa (ISSA), pp. 1–8, Johannesburg, South Africa, August 2011.

[14] S. Yadav and A. L. N. Reddy, "Winning with DNS failures: strategies for faster botnet detection," in Proceedings of the 7th International ICST Conference on Security and Privacy in Communication Networks (SecureComm), London, UK, September 2011.

[15] E. Stinson and J. C. Mitchell, "Towards systematic evaluation of the evadability of bot/botnet detection methods," in Proceedings of the 2nd USENIX Workshop on Offensive Technologies (WOOT), San Jose, CA, USA, July 2008.

[16] Alexa Top Global Sites, http://www.alexa.com/topsites

[17] J. L. Myers and A. D. Well, Research Design and Statistical Analysis, New York, NY: Lawrence Erlbaum Associates, 2003.

[18] P. N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Boston, MA: Addison-Wesley, 2005.

[19] Open Malware - Community Malicious Code Research and Analysis, http://www.offensivecomputing.net

978-1-4799-2093-8/13/$31.00 ©2013 IEEE