PsyBoG: Power Spectral Density Analysis for Detecting Botnet Groups

Jonghoon Kwon, Jeongsik Kim, Jehyun Lee, Heejo Lee[§]

Dept. of Computer Science and Engineering
Korea University
Seoul, Republic of Korea
Email: {signalnine, misia, arondit, heejo}@korea.ac.kr

Adrian Perrig

Institute of Information Security
ETH Zurich
Zurich, Switzerland
Email: adrian.perrig@inf.ethz.ch

***Abstract*—Botnets are widely used for acquiring economic profits, by launching attacks such as distributed denial-of-service (DDoS), identification theft, ad-ware installation, mass spamming, and click frauds. Many approaches have been proposed to detect botnet, which rely on end-host installations or operate on network traffic with deep packet inspection. They have limitations for detecting botnets which use evasion techniques such as packet encryption, fast flux, dynamic DNS and DGA. Sporadic botnet behavior caused by disconnecting the power of system or botnet's own nature also brings unignorable false detection. Furthermore, normal user's traffic causes a lot of false alarms. In this paper, we propose a novel approach called PsyBoG to detect botnets by capturing periodic activities. PsyBoG leverages signal processing techniques, PSD (Power Spectral Density) analysis, to discover the major frequencies from the periodic DNS queries of botnets. The PSD analysis allows us to detect sophisticated botnets irrespective of their evasion techniques, sporadic behavior and even the noise traffic generated by normal users. To evaluate PsyBoG, we utilize the real-world DNS traces collected from a /16 campus network including more than 48,046K queries, 34K distinct IP addresses and 146K domains. Finally, PsyBoG caught 19 unknown and 6 known botnet groups with 0.1% false positives.**

***Index Terms*—Botnet detection, Power Spectral Density, Group Activity**

I. Introduction

A bot is a malicious program remotely controlled by an attacker known as a botmaster. A botnet is a network of bot-infected computers. Botnets are widely used for acquiring economic profits, by launching attacks such as distributed denial-of-service (DDoS), identification theft, ad-ware installation, mass spamming, and click frauds. A McAfee threat report [18] foresees that botnet activities continuously increase in 2013. The report also stresses the fact that new and unseen types of botnets are continuously being developed. Also, the botmaster uses smart devices such as smart phones and smart TVs as their instruments to commit the cyber crimes. As botnets show high infection rates and are capable of conducting wide range network attacks, the botnets are considered as a major threat to cyber security.

Looking into botnets from various angles, several characteristics can be recognized. Many methods to detect botnets using the characteristics have been suggested, and they can be classified into two categories: host-based detection and network-based detection. The host-based detection method mainly aims to analyze the internals of a computer system [23] [15]. The host-based detection method allows users to detect abnormal activities in the system easily. However, the host-based method has limitations such as its difficulty to apply to a wide coverage of hosts due to the overhead while trying to detect the bots.

The network-based detection method monitors the network traffic at servers and routers. It does not have the issue with a wide coverage of hosts. Early researchers focused on the contents of a botnet [22] [7]. However, as the botnets apply the techniques of encryption and obfuscation, detecting botnet with the content-based methods became difficult. Traffic pattern-based detection was suggested as an alternative. It analyzes the pattern of network traffic generated by the infected host [8] [6]. However, the traffic-pattern based detection has suffered from low detection rate caused by normal user's traffic. The ***sizable volume*** of the traffic generated in the network is a significant issue as well. In addition, caused by disconnecting the power of system or botnet's own nature, botnet behavior could be very sporadic which makes the detection more difficult.

Some botnet detection researches focused on specific network traffic. *Choi et. al.* proposed a botnet domain detection mechanism by monitoring DNS traffic [5]. The mechanism searches groups of computers that make DNS queries periodically to the same domain during a certain period of time. Unfortunately, botnet authors evade the detection using multiple domains randomly like fast flux. In fact, DGAs (Domain Generation Algorithms) [20] [2] are widely used in various bot codes for bypassing existing domain-based detection systems (i.e., domain blacklist) [4]. DGAs periodically generate a large number of domain names that can be used as rendezvous points between the bot hosts and C&C servers, thus the domain name based analysis is not sufficient to response against the botnet threats anymore.

In this paper, we propose a novel network based approach called PsyBoG to detect botnet groups by capturing intrinsic natures of botnet. At first, we observe periodic behaviors of botnets. More precisely, bot hosts communicate with C&C (Command and Control) servers or other bot hosts regularly for maintaining availability and capability of the botnet in peacetime. Even when the botnets launch attacks such as mass spamming, click frauds and DDoS, each attack causes huge and regular network traffic. Those activities have the periodicity of botnet behaviors which is one of the unalterable

[§] Corresponding author: heejo@korea.ac.kr

978-1-4799-7329-3/14/$31.00

characteristics. PsyBoG leverages PSD (Power Spectral Density) analysis which is one of signal processing techniques to discover the major frequencies from those periodic behaviors of botnet. The PSD analysis solves the problems of sporadic behaviors of botnet and normal user generated traffic acting as a determent.

Secondly, PsyBoG discovers a group activity of botnets. A botnet is a group of the compromised machines controlled by an attacker, the so called a botmaster. Driven by this, we can easily expect that the bot hosts exhibit similar behavior patterns. More sophisticated botmaster can divide the entire bot hosts into multiple subgroups. However bot hosts from the same subgroup still show similar behavior patterns, since the fundamental aim of botnet is to commit the cyber crimes which can only be done by huge amount of resources. We define the similar behavior patterns as a group activity which is an inherent property of the botnet. PsyBoG groups hosts in accordance with the similarity of traffic patterns.

PsyBoG analyzes DNS traffic only. Sizable volume of current Internet traffic is a significant issue, and the traffic volume skyrockets nowadays. DNS traffic is an alternative to be considered in terms of practical use of botnet defense mechanism since DNS is an important part of botnet life-cycle. PsyBoG does not rely on the domain names in DNS traffic, unlike previous researches or existing countermeasures such as DNS sinkhole and blacklist filtering, but the periodicity of DNS query pattern typically shown within botnet traffic. The problem of traffic encryption and obfuscation, thus, is no longer an issue, and neither are fast flux, dynamic DNS and DGA.

In experimental results with real campus DNS traces, PsyBoG shows high detection rate without prior knowledge. PsyBoG discovers not only 6 known botnets, but also 19 unknown botnets which are not listed in the latest blacklists. Furthermore, only 0.1% of IP addresses were considered as the false positives.

In summary, this paper has following contributions:

- We suggest a viable approach that is resilient against the noise traffic generated by normal users.
- Our mechanism does not require any prior knowledge of botnets, such as the binary signatures, traffic signatures and training data.
- Our mechanism is an effective countermeasure against sophisticated botnets which utilize evasion techniques, such as payload encryption, frequent change of C&C communication pattern, fast flux, and DGAs.

The rest of the paper is organized as follows. Section 2 describes the background of our mechanism including the characteristics of botnet behaviors and PSD analysis. Section 3 describes the problem statement and requirements for botnet detection. Section 4 introduces our mechanism, and section 5 outlines the experimental settings and results. In section 6, we discuss potential techniques that may exploit our mechanism. Section 7 describes related works, and finally we conclude this work with outline of future works in section 8.

II. Background

We briefly introduce the background knowledge of our work: Concept of botnet and signal processing techniques.

A. Concept of botnet

We use three inherent features of botnets for detection: utilization of DNS, periodic communication and group activity of bot hosts. Detailed explanations are as follow.

*1) **Utilization of DNS**:* Bot infected hosts use DNS to access the C&C servers. Because botnet authors have suffered from disclosure of IP addresses for C&C servers by reverse engineering, they attempt to use domain names instead of static IP addresses. Furthermore, they prevent detecting and blocking of the servers through periodically changing the C&C server IPs and domain names such as fast flux or DDNS (Dynamic DNS).

Generally, a bot transmits DNS queries to keep connection with the C&C servers. More intelligent botnets use DDNS to hide their query pattern. A C&C server using DDNS changes its IP address frequently and has smaller TTL (time to live) values in its DNS record, resulting in more frequent DNS queries from the bots. However, in the case of using data stored in the local cache, the DNS query is invisible as it is not transmitted outwards.

*2) **Periodic communication**:* A bot program is set to communicate periodically with the C&C server. This is because the connection with the C&C channel is needed for the C&C server to check the host status or issue an attack. The periodic connection guarantees availability and capability of the botnet. A bot-infected machine automatically accesses the C&C server of the botnet. The bot host queries predefined domain names to a DNS server to obtain the IP address of the C&C server and periodically reports its status to the C&C server. More intelligent botnets change their IP addresses from time to time using DDNS. In addition, a DDNS server maintains the small TTL values. In this case, more frequent DNS queries can be observed.

*3) **Group activity**:* Botnet communications can be observed as the form of group activities. Botnet needs certain rules which are previously defined to manage several hundreds, or even thousands of bot hosts. A centralized botnet (IRC, HTTP) uses DNS to look up the C&C server. The botnet sends periodic DNS queries and connects to the channel. This DNS lookup is an good example of the group activity.

A distributed (P2P) botnet performs group activities which are observed while the P2P botnet performs upgrade or synchronization. For example, the storm P2P botnet often synchronizes with the network time protocol server through the infected host. This synchronization activity also shows a group activity. When a botnet performs malicious behaviors including DDoS attack, spamming and click frauds, each bot host generates massive attack traffics simultaneously for more efficient and effective attack. For example, in DDoS attack, numerous bot hosts must launch attacks to target systems concurrently.

B. Signal Processing Techniques

PsyBoG utilizes a signal processing technique for extracting the periodic communication pattern in a botnet. Signal processing is the operations for the analysis of analog and digitized signals. One of the typical operations in signal processing is to extract frequencies from a given sequence of signals.

A DFT converts a discrete-time domain signal such as time series to a frequency domain data as a sum of sinusoidal

components (*sine* and *cosine*). The frequency domain data have amplitudes of each frequency. A fast Fourier Transform (FFT) is an efficient algorithm which can conduct a discrete Fourier transform (DFT) and its reverse execution in a short time. While the time complexity of DFT is O(N^2), FFT shows a time complexity of O($Nlog_2N$).

When we input the sequence of data points (time series) f(1), f(1), f(2), ..., f(N) to Equation 1. N (2^n) is the size of the entire data, *n* is the index value, and k is the frequency which needs to be known. The transform to frequency area result shows complex numbers of $F(1)$, $F(1)$, $F(2)$, ..., $F(N)$.

$$F_N = \sum_{n=1}^{N} f_n \cdot e^{-i2\pi kn/N}. \quad (1)$$

$$P_{xx}(\omega) = \frac{(\Delta t)^2}{T} \left| \sum_{n=1}^{N} f_n e^{-i\omega n} \right|^2 . \quad (2)$$

We assume that the periodic pattern of host traffic is figured out from high amplitudes of certain frequencies. Second, Equation 2 is the definition of the power spectral density (PSD). PSD generalizes in a straightforward manner to finite time-series f_n with $0 \leq n \leq N$, such as a signal sampled at discrete times $f_n = f(n\Delta t)$ for a total measurement period $T = N\Delta t$. $P_{xx}(\omega)$ is the average of the Fourier transform magnitude squared and ω is $2\pi k/N$. The PSD describes how the power of a signal or time series is distributed with an unit of energy per frequency. The power can be defined as the squared value of the signal.

III. Problem definition

We describe the problem when detecting botnets. Then we suggest the requirements for solving the problem along with the goal of our study.

A. Problem Statement

We state following three problems for botnet detection.

*1) **The huge network traffic volume**:* As the Internet traffic skyrockets, the previous anomaly detection mechanisms require more resources and times to analyze the entire network traffic for enough investigation. Also, false negative and false positive of detection can be occurred when scanning a mass volume of network traffic. Therefore, massive resource consumption, false negative and false positive of detection are the issues for the network-based method nowadays.

*2) **Insufficient detection**:* Detecting one or some of the bot hosts is insufficient to prevent the threat of botnet. A botnet is a group of multiple hosts, so it is difficult to prevent attacks from the botnet without blocking the host group. Detecting a subset of bot host group is not effective in preventing botnet attacks.

*3) **Overhead from covering a wide network range**:* The bigger network range we try to cover, the more hosts we need to consider. As the number of hosts increases, the traffic needs multiple scanning, and leads to high false alarm ratio and long computation time.

B. Requirements

Previous studies did not provide effective solutions to the problems. The following requirements are addressed to overcome the problems above. The following requirements must be met to solve the problems we have defined.

*1) **Low volume traffic**:* Motivated by situations in which the traffic volume increases rapidly, only low volume traffic which is closely associated with the botnet must be analyzed, instead of analyzing the entire network traffic.

*2) **Botnet group detection**:* Botnet groups must be detected through their group activities. Attacks of a botnet can be prevented from blocking the botnet groups. To do so, botnet groups must be detected.

*3) **Efficient traffic analysis**:* The traffic overhead must be controlled. As the number of hosts increases, the amount of traffic to be analyzed also increases. Therefore, a new method needs to monitor the traffic and analyze them more efficiently.

C. Goal

Our goal is to suggest a method which can detect botnet groups by scanning the periodic pattern of traffic without any prior knowledge of botnet, in an efficient manner using only a small portion of traffic.

IV. Proposed Mechanism

In this section, we introduce the concept and structure of our mechanism which is called PsyBoG (Power Spectrum analYsis for detecting Botnet Groups). Then, we explain the operations of PsyBoG in details.

A. Overview

1) Detection method concept: The periodicity of a bot host can be extracted from the DNS traffic using PSD. Botnet groups can be detected by performing a similarity measurement from the periodicity of bot hosts.

2) Detection method structure: Figure 1 shows the structure of PsyBoG. The structure consists of four modules.

- **Traffic collector.** Collects DNS traffic such as host IPs, domain names and query time stamps from the DNS servers.
- **Host periodicity analyzer.** Uses PSD to extract the frequency information of host DNS traffic.
- **Significance peak analyzer.** Analyzes the significance of peak values in a power spectrum. If a peak crosses the significance threshold, PsyBoG determines that the host contains very suspicious periodic query patterns.
- **Group activity analyzer.** Analyzes the similarities of the power spectrums between the hosts, which contain a significant periodic component. If the power spectrums show a high similarity rate with each other, they have similar periodic query patterns and belong in the same botnet group.

B. DNS Traffic Collection

The sensors collect the DNS traffic of a monitored network by tapping DNS servers and aggregate the DNS traffic to the DNS traffic collector. One problem is the huge amount of network traffic. To seize this problem, we only use the DNS traffic that has a smaller volume but closely relates with

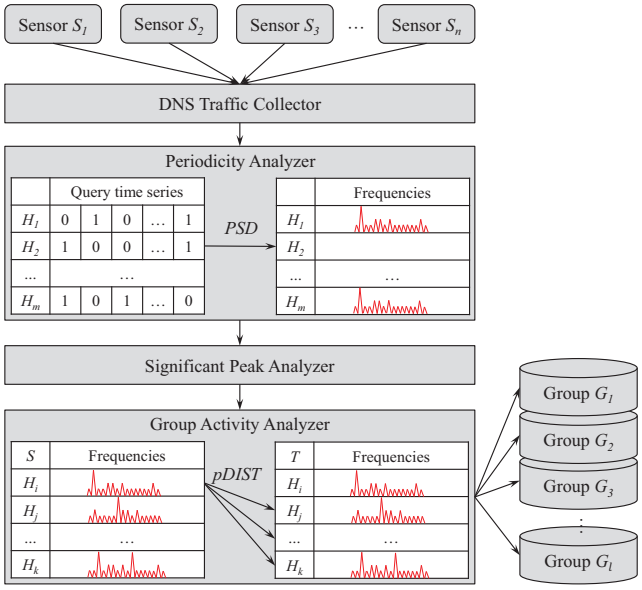

Fig. 1: Design of PsyBoG architecture.

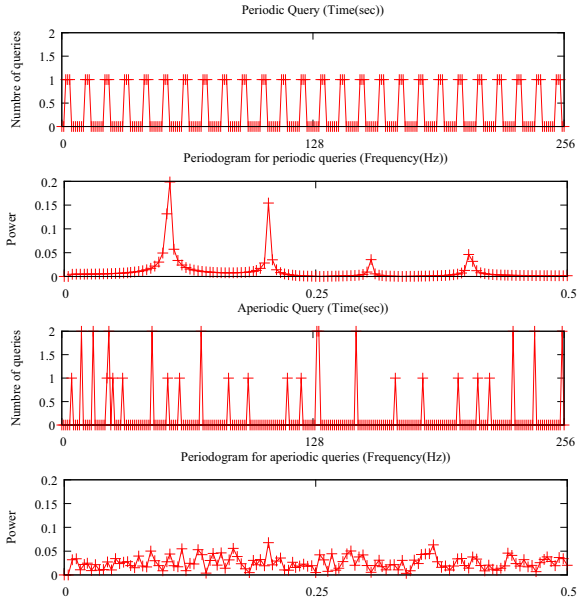

Fig. 2: Periodograms for periodic and aperiodic signals.

botnet activities. Moreover, numerous botnets apply the highly advanced DNS techniques such as fast flux, DDNS, and DGAs to avoid detection of C&C and spam servers, so they show a higher rate of DNS usage compared to normal users or systems.

The malicious activities of the botnet can be prevented by blocking the DNS traffic. As the C&C server periodically changes its IP address, blocking the DNS query of botnet hosts disables the hosts from accessing the server as they do not know its address. However, this may result in the problem of DNS caching. The DNS cache can differ from operating systems or browsers. Therefore, cache of different kinds of operating systems and browsers must be taken into account. Meanwhile, the advanced DNS techniques that are used by the newer botnets keep a shorter TTL, thus the DNS caching problem can not occur.

To make the use of DNS traffic, we extract the host IP which sends the domain request, the name of the domain, and its time stamp from the aggregated DNS traffic.

C. PSD Analysis

We can analyze the periodicity of the botnet communications with PSD. We assume that a periodic traffic of botnet is transformed into certain frequencies with high power in PSD, while an aperiodic traffic caused by normal users is transformed into low power over broad frequencies in PSD. Therefore, botnet traffic can be extracted by scanning for certain frequencies with high-energy. We can transform signal-time data into the frequency data using PSD.

First, we set a number of segments of input time series for operating PSD. The number of segments affects a frequency domain and its range. For a high-quality PSD, the number of segments is selected among powers of two, and we limit the length of the number of segments to 2^{14} = 16,384 for a fast PSD analysis. Note that, the size of single segment is an one second, we apply the sliding time-window strategy to cover a long input trace. The second step in PSD estimation is to remove the mean value of the Fourier mode from the time series. This is a standard technique [21] that allows more accurate PSD estimation. In the third step, we use a Hanning window that is used on half-overlapped intervals for best Signal to Noise Ratio (SNR). The last step is operating PSD analysis for the segments of input time series.

Fig. 2 represents an example for the periodic and aperiodic signals and their periodograms which project the corresponding PSDs for the signals respectively. The first plot shows a periodic query sample in which a period of query is 10 sec, a duration of each query is 3 sec, and a number of segments is 256. The second plot that is a periodogram for the first plot, consists of a largest peak at 0.1Hz, small peaks at multiples of the largest peak 0.1 Hz, and almost zero, which indicates that the original signal has a periodic query pattern within every 10 sec. The third plot contains an aperiodic signal that follows the Gaussian random distribution, and it's associated periodogram, the last plot, contains several peaks but none of them has sufficiently large enough power, thus there is no periodic pattern. Note that the decision for whether a peak value is large enough or not, will be explained in following section.

D. Significant Peak Testing

As we can see in Fig. 2, there is always a peak in the periodogram irrespective of whether the original signal contains a periodic pattern or not. Therefore, the decision for whether the peak is significant enough to determine that the peak is caused by a periodic component with a specific frequency or not, have to be issued. We apply the significant testing of periodogram ordinates referred from previous studies [3] [11] [14].

The significant testing for periodogram ordinates is designed to work on the binary hypothesis. For a query sequence $x[n]$,

H_0 : x[n] follows Gaussian distribution, while

H_1 : x[n] has a periodic component at the largest ordinate.

For H_0, the ordinates $P_{xx}[\omega]$ for the sequence $x[n]$ has a distribution which is proportional to χ^2 in two degrees of freedom. Therefore,

$$P_{xx}[\omega] = \sigma_x^2 \chi_2^2 \tag{3}$$

The probability distribution of a χ_2^2 is an exponential function [12],

$$f(z) = 2^{-1} exp(-z/2) \tag{4}$$

Hence, for any value of $z \geq 0$, the probability that $P_{xx}[\omega]/\sigma_x^2 \leq z$ is given by,

$$\begin{aligned} Pr[P_{xx}[\omega]/\sigma_x^2 \leq z] &= \int_0^z f(x)dx \\ &= \int_0^z 2^{-1} exp(-x/2)dx \\ &= 1 - exp(-z/2). \end{aligned} \tag{5}$$

Under H_0 that γ_x indicates one of the $N/2$ independently identically distributed variables, then for any value of $z \geq 0$,

$$\begin{aligned} Pr[\gamma_x > z] &= 1 - Pr[P_{xx}[\omega]/\sigma_x^2 \leq z, \text{for}\, \omega] \\ &= 1 - [1 - exp(-z/2)]^{N/2} \end{aligned} \tag{6}$$

Eq. 6 can be used for determining whether or not the largest ordinate in the periodogram is significantly different from a zero mean distribution with variance σ_x^2. The variance σ_x^2 can be evaluated directly,

$$\sigma_x^2 = N^{-1} \sum_{k=1}^{N/2} P_{xx}[\omega] \tag{7}$$

According to above estimation of σ_x^2, we can drive g_x* from Eq. 3,

$$g_x^* = \frac{max(P_{xx}[k])}{N^{-1} \sum_{k=1}^{N/2} P_{xx}[k]} \tag{8}$$

Under H_0, g_x^* will have the same distribution as γ_x, thus for $z \geq 0$,

$$Pr[g_x^* > z] \sim 1 - [1 - exp(-z/2)]^{N/2} \tag{9}$$

The testing the significance of the periodogram peak is widely used in numerous researches, and PsyBoG also applies this testing to figure out the hosts which generate DNS queries periodically. By simply applying this testing to the example exhibited in Fig. 2, we can get the following results. The threshold $z_{0.1\%}$ with the binary hypothesis for the expecting false positive rate 0.1% is 23.52, where the number of segments $N = 256$. The periodic and aperiodic signal(first and third plot) return g_x^* as 35.34 and 5.39 respectively. From this, we can determine the first plot has a periodic component at frequency $k = 0.1Hz$ but the third plot has not.

E. Botnet Group Activity Detection

PsyBoG investigates the botnet groups since detecting the groups is necessary for efficient prevention of botnet threats. To do this, we apply the similarity measurement algorithm pDist (Power Distance) [25] to detect the group activities of botnet. pDist compares the periodic structure of two input signals. More precisely, pDist utilizes the specific frequencies in the periodogram, which contains the large enough power mentioned above. Lets assume that there are two distinct periodograms $P_{aa}[k_n]$, $P_{bb}[k_n]$ with length n, and the periodograms are discovered that their largest ordinates exceed the threshold for the significant testing. Now, we can get the frequencies with the large enough power values $p_a \subset [\{x_1, y_1\}...\{x_i, y_j\}]$. Finally, we can simply compare the power values located at the frequencies f_a with f_b. The distance pDist represents the similarity between two signals a and b,

$$pDist = \|f_a - f_b\| \tag{10}$$

In [25], they described an experiment about the clustering accuracy for pDist with four other clustering approaches such as Euclidean, DTW, Cepstrum and CDM. Among them, pDist shows not only the highest accuracy but also lightweight, since pDist works on a very low dimensional space, precisely only for the i dimensions.

V. Experimental Results

To evaluate the performance of PsyBoG, we collect the real-world DNS traces. The DNS traces are obtained by tapping from the gateway router of a /16 campus network on Feb. 24th, and 25th, 2014. We can give a brief overview of the traces. The first day trace (Feb. 24th) contains total 24,278K queries generated by 25,647 IPs for 100,516 domains, and the second day trace (Feb. 25th) contains 23,768K queries by 25,300 IPs against 93,350 domains.

A. Preprocessing for data set

1) Filtering: We apply both of host and domain filtering for effective DNS traffic analysis. Even though the DNS traces contain at most 65K IPs, analyzing every host, including similarity calculation, is a task consuming too much time. Hence, we decide to remove the hosts that have DNS queries less than 5 in one day DNS traffic because it is too small to discover some periodicity.

Domain queries listed on a whitelist are also excluded from our trace. Recently, many benign programs constantly connect to specific domains to make their service concrete, i.e., Windows update, and AV update. Furthermore, super famous Web sites like Google, Facebook, and Twitter are queried very frequently. These well known and thriving domains would therefore expose some periodicity. To this end, we built the whitelist including Top 500 domains collected from Alexa.com [1].

2) Time series: In this step, we sample the DNS trace like binary signal by assigning it to be 1 at each query request and 0 at intervals of query requests, and a sampling interval is a second. According to the recommendation for accurate and fast signal processing operation, power of two, precisely 256, 512, 1,024, until 16,384, is regarded as the length of time

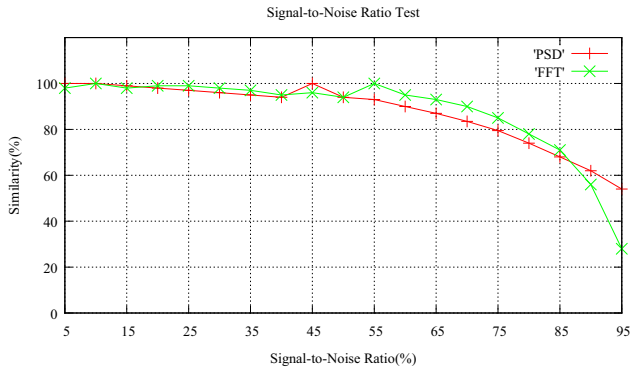

Fig. 3: The similarity rate of botnet traffic mixed noise with naive botnet traffic.

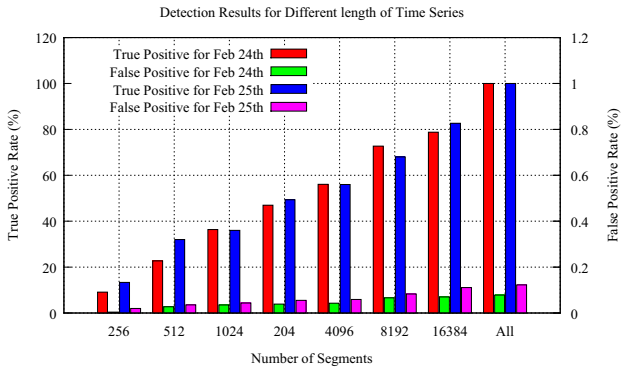

Fig. 4: The detection accuracy as the number of segments of time series.

series. We observed the introduction of large gaps between queries, but the large gap does not significantly compromise the PSD results [24].

B. Signal to Noise Test

SNR (Signal-to-Noise Ratio) has been researched in previous studies, and it already proves that the noise signals cannot influence other spectrum components in frequency domain. Nevertheless, we conduct an experiment on similarity analysis according to SNR to investigate that PsyBoG is robust to the user noise. The reason is that there is no guarantee that the similarity for the significant ordinates of periodogram between two signals would not be affected by noise, even though the noise has no effect to the ordinates where the periodicity located in. In this experiment, we utilize the actual bot traffic with 51 second periodicity, and then we randomly add noise traffic by increasing SNR.

Fig. 3 shows the variation of similarity according to user traffic. we perform the experiment using PSD and FFT. Y and X axis represent the similarity ratio and the SNR respectively. When the noise traffic accounted for 75% of the total traffic, pDist between the naive bot traffic and the mixed traffic archives around 80% of similarity. Over 75% of SNR, pDist still shows reasonable similarity results, thus we can say that PsyBoG has an ability to discover the periodic behavior and group activities of the botnet, even if there are a lot of user queries in the DNS traces.

C. Detection Accuracy

As aforementioned, the input data stands on the different length of time series. The difference gives impact on the detection accuracy, thus we investigated the relations between the number of segments and the detection accuracy to figure out the most efficient length of time series for input data. Fig. 4 exhibits the detection results for the different length of time series. The results show that increasing the number of segments brings better true positive, while it holds false positive around 0.1%.

Interestingly, even though small number of segments (i.e., 256, 512, and 1024) shows low true positives, the detection results are still valuable since some bot hosts have been discovered only in the input settings which have small number of segments. According to our analysis, this result was caused by the sporadic behavior of the bot hosts. For example, a bot host operates only for a while due to the system turning off by a host owner. In such a case, even if the bot host shows the periodic behavior, it only can be reflected in a short period of time series. Therefore, in a large size of time series, the periodic behavior is not able to affect as a significant peak. From the fact, we decided to perform the experiments with different lengths of time series.

D. Botnet Group Detection

Table I represents the final detection result for the two days DNS traces. Note that we applied DNS blacklist [17] as a ground-truth for classifying the detection result into known botnet, unknown botnet, and false positives. If a domain name is involved in the periodic DNS queries and the domain was listed on the blacklist, we determined the host group which queried the domain as a known botnet. If a domain name was queried periodically but was not listed on the blacklist, then we manually investigated through Google search to determine whether the host group is an unknown botnet or false positive.

As a result, PsyBoG discovered 6 known botnets, 19 unknown botnets, 4 adwares, 3 suspicious groups and 5 false positives. The suspicious groups showed relatively high periodicity and abnormal domain queries. Especially, one group generated queries against more than 3,000 domain names which are variants of 15 original domains. The domain variations showed very similar patterns with the domains generated by DGA. The only reason why we classified the groups as suspicious is that there was no proper evidence to prove that the domains were malicious. The other false positives were caused by legitimate services such as Torrent trackers, mail delivering services, and NTP. One false positive group was discovered as a scientific research crawler which crawled some information from thousands of Web servers. Despite of the low false positive rate of PsyBoG (0.1%), we expect that the false positive rate is still able to be reduced by listing the legitimate domains in the whitelist.

VI. Security Analysis

In this section, we discuss potential techniques that may aim to exploit PsyBoG, and discuss how PsyBoG is resilient against the techniques.

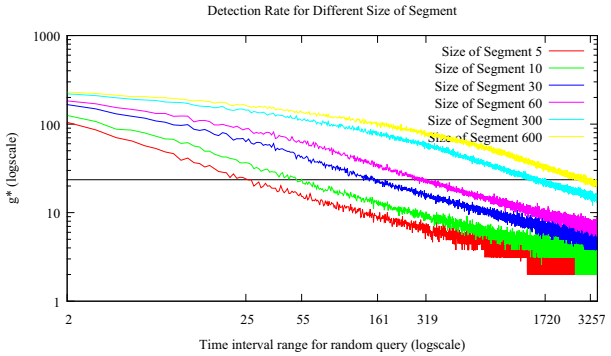

Fig. 5: Periodicity measurements against randomly generated query pattern.

A. Random Query Pattern

PsyBoG leverages the periodic query patterns of bot program in accordance with the assumption that the bot program simply follows the source code that is written for making the bot operates automatically and constantly. We already showed that the assumption holds for the real-world botnet through the experiments. Nevertheless, bot writers might attempt to evade PsyBoG by modifying their query patterns. A reasonable potential alternative is applying random function to generate non-periodic queries. More precisely, bot authors can randomize the time interval between DNS queries by applying random function $rand(y)$, where 0 to y is the time interval range. The randomized time interval i located between range 0 to y, could exploit the periodicity of query patterns. Interestingly, PsyBoG, more specifically PSD analysis, already has resilience to the randomization.

To address this, we analyzed the periodicity measurements against artificially randomized query patterns. First, we generated random query traffics with different maximum random ranges from 2 to 3,600, which means that, i.e., a random query traffic $x[n]^{rand(600)}$ has a query sequence such as x_{k-1}, x_k, and x_{k+1}, and the time intervals between the queries can be random numbers between 0 and 600. Second, the input time series was built with different size of segment s such as 5, 10, 30, 60, 300, and 600 seconds. According to s, every query generated on time from t to $t+s$, is accumulated into a single segment of time series T_j. Finally, PSD of Time series T was analyzed. Fig. 5 depicts the significant peak testing with the randomized query patterns.

When s is 5 sec, $g*$ has exceeded the threshold of significant peek testing $z_{0.1\%}$ for the random query traffics $x[n]^{rand(25)}$. Furthermore, the testing showed that more complicated random query patterns which have maximum random range 3,257 were discovered by increasing s to 600 sec. As a result, the simple experiment represents that PsyBoG has a resilience against the random query patterns. Of course, the changes of the segment size may bring the increase of false positives. We remain the problem as a future work.

B. Slow Query Pattern

Bot authors might apply slow query patterns to hide their communications. For example, some bot can generate DNS query once in an hour, a day, or a week to communicate with the botmaster. However, in case of applying slow query patterns, it causes a serious deterioration of function of botnet.

TABLE I: Detection results for botnet group

Type	Botnet Name	Domain Name	IP#
Known Botnet	Palavo	ilo.brendz.pl ant.trenz.pl	2
	Palavo(2)	peer.pickeklosarske.ru juice.losmibracala.org jebena.ananikolic.su teske.pornicarke.com	3
	Palavo(3)	pica.banjalucke-ljepotice.ru sandra.prichaonica.com l33t.brand-clothes.net	2
	W32/Tupym-D	h1.ripway.com	3
	Worm.Win32.AutoRun.fnc	www.balu0{xx}.0catch.com (14)[1] www.gearext.com	
	Trojan-pws.Win32.QQPass	{xxx(xxx)}.{xx(x)}.ijinshan.com (4) {xx(x)}.{xx(xxx)}.duda.net (14) up.liebao.cn	6
Unknown Botnet	N/A	{xx}.cdn.qhimg.com (4) {xxx}.shouji.360tpcdn.com (3)	24
		loveting.no-ip.org	1
		sexman69.mlbfan.org	3
		sh.rstrainer.net sh.rstrainer.net.local	1
		biz5.sandai.net miserupdate.aliyun.com	9
		www.xiaopijia.com	2
		data2.168sm.com	1
		r.usyncapp.com t.usyncapp.com	12
		l33t.brand-clothes.net	3
		d2uzsrnmmf6tds.cloudfront.net	92
		{xxx(xxx)}orbitdownloader.com (3)	2
		sc{xx}.rules.mailshell.net (6)	10
		js.moatads.com	45
		optimizedby.brealtime.com	35
		mydati.com	5
		c.ztstatic.com	5
		cm1.jssearch.net	109
		s{x}.kgridhub.net (6)	37
		hcimg.realclick.co.kr	51
	Adware	servedby.bigfineads.com	2
		b.scorecardresearch.com	61
		servedby.bigfineads.com	2
		servedby.myinfotopia.com	4
Suspicious Group	N/A	{x(xxxxxxx)}.www.0538hj.com (2644) {x(xxxxxxx)}.lieb.76yxw.com (40) {x(xxxxxxx)}.www.8885ok.com (82) {x(xxxxxxx)}.baidu.915hao.com (178) {x(xxxxxxx)}.www.jrj001.net (38) {x(xxxxxxx)}.www.kr5b.net (40) {x(xxxxxxx)}.wwee.iinfobook.net (154) {x(xxxxxxx)}.www.jeeweb.net (40) {xxxxxxxx}.www.chuansf-1.com (19) {x}.wwww.boeeo.com (10) {x}.www.booooook.com (24) {x}.www.jiaduolu.net (15) {x}.jiaduolu.net (15) {x}.qianliri.com (13) {x}.hj19.com (24)	1
		{xxxx(xxxx)}.com (134) {xxxxxxxx}.info (128) {xxxxxxxx}.net (132)	3
		xayazkesh.in dubstepdrop.net halo4beta.co	6
False Positive	Torrent tracker	i.e., tracker.gaytorrent.ru	6
	Crawler	N/A	3
	Mail server	N/A	1
	NTP	N/A	3
	Etc.	N/A	24

[1]The domains have (n) variations with changing characters at the braces.

The only reason of the rampant of botnet is that botnet provides an ability of concentration of attack resources for massive attacks, therefore the power of botnet totally depends on its availability and capability. But the slow query pattern ruins the availability and capability, since slow query limits the response for the master's commands. Generating slow queries limits the agility of botnet as well, hence it may cause the single point failure. Despite of the limitations, botmaster can apply the slow query generation. Fortunately, PSD analysis is not affected by the large gap of two queries. As we can see the Fig. 5, PSD analysis shows significant enough peak values against large gap of queries (almost 1 hour of time interval). Therefore, the slow query generation is not an issue any longer.

VII. Related work

Recently, many researchers have proposed new botnet detection methods, and some of them are host based methods. BotSwat [23] traces all input data using a taint propagation trace technique to uncovered botnet commands. Unfortunately, BotSwat showed unignorable false alarms and high system overheads due to the taint propagation. BotTracer [16] monitors three phases of botnets on the virtual machine, but it also has high false alarms since the three phases of behavior can be monitored to normal program too. Jacob et al. present JACKSTRAWS [13], which analyzes botnet binaries by monitoring system call graphs for C&C communication. However, recent botnets may not strictly follow the system call graphs, therefore JACKSTRAWS can not be a fundamental solution.

There have been efforts on network-based detection such as Bothunter, BotSniffer, BotMiner, and BotGrep. BotHunter [9] models a botnet infection model and deals with IDS-driven dialog correlation. BotSniffer [10] focuses on a highly synchronized communication of botnets, and BotMiner [8] applies clustering algorithms to perform cross-plane correlation. BotGrep [19] analyzes C&C communication on the overlay topologies to defeat P2P botnets. In spite of the outstanding researches, the network-based botnet detection is still suffered from high false alarms and significant overhead due to the massive size of traffic volumes. Tegeler et.al. proposed BotFinder [24], a system to detect infected hosts in a network using only high-level properties of bot traffic. BotFinder applies a clustering approach to model botnet behaviors, especially bot traffic patterns including time interval, duration, and FFT of communication. This work is the one which is partially similar to our work, but BotFinder still suffers from the user generated traffic and huge volume of network traffic.

VIII. Conclusion

In this paper, we propose a novel botnet detection approach to detect activities of the latest botnets. The approach solves several problems including that user's traffic acted as a noise for botnet detection, variable communication time between bot host and C&C server, and evasion techniques such as a payload encryption, fast flux, and DGAs. In future works, we will adopt finer grained clustering methods to figure out malicious domain lists automatically, and evaluate the performance of PsyBoG with a larger data set.

IX. Acknowledgement

This work was supported by the ICT R&D program of MSIP/IITP. [14-824-06-001, The Development of Cyber Blackbox and Integrated Security Analysis Technology for Proactive and Reactive Cyber Incident Response]

References

[1] Alexa, "Alexa top 500 sites on the Web," 2014, http://www.alexa.com.
[2] M. Antonakakis, J. Demar, C. Elisan, and J. Jerrim, "Dgas and cybercriminals: A case study," 2012.
[3] B. AsSadhan, J. M. Moura, and D. Lapsley, "Periodic behavior in botnet command and control channels traffic," in *IEEE Global Telecommunications Conf., GLOBECOM*, 2009, pp. 1–6.
[4] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "Exposure: Finding malicious domains using passive dns analysis." in *NDSS*, 2011.
[5] H. Choi, H. Lee, and H. Kim, "Botgad: detecting botnets by capturing group activities in network traffic," in *Proc. of the Fourth Int'l ICST Conf. on COMmunication System softWAre and middlewaRE, COMSWARE*, 2009, p. 2.
[6] C. J. Dietrich, C. Rossow, and N. Pohlmann, "Cocospot: Clustering and recognizing botnet command and control channels using traffic analysis," *Computer Networks*, 2012.
[7] J. Goebel and T. Holz, "Rishi: Identify bot contaminated hosts by irc nickname evaluation," in *Proc. of the First Conf. on Hot Topics in Understanding Botnets*, 2007, pp. 8–8.
[8] G. Gu, R. Perdisci, J. Zhang, W. Lee *et al.*, "Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection." in *USENIX Security Symposium*, 2008, pp. 139–154.
[9] G. Gu, P. A. Porras, V. Yegneswaran, M. W. Fong, and W. Lee, "Bothunter: Detecting malware infection through ids-driven dialog correlation." in *USENIX Security Symposium*, 2007, pp. 1–16.
[10] G. Gu, J. Zhang, and W. Lee, "Botsniffer: Detecting botnet command and control channels in network traffic," in *NDSS*, 2008.
[11] G. Hernandez, "Time series, periodograms, and significance," *Journal of Geophysical Research: Space Physics (1978–2012)*, vol. 104, no. A5, pp. 10 355–10 368, 1999.
[12] P. G. Hoel *et al.*, "Introduction to mathematical statistics." *Introduction to mathematical statistics.*, no. 2nd Ed, 1954.
[13] G. Jacob, R. Hund, C. Kruegel, and T. Holz, "Jackstraws: Picking command and control connections from bot traffic." in *USENIX Security Symposium*, 2011.
[14] C. Koen, "Significance testing of periodogram ordinates," *The Astrophysical Journal*, vol. 348, pp. 700–702, 1990.
[15] J. Kwon, J. Lee, and H. Lee, "Hidden bot detection by tracing non-human generated traffic at the zombie host," in *Proc. of Int'l Conf. on Information Security Practice and Experience*, 2011, pp. 343–361.
[16] L. Liu, S. Chen, G. Yan, and Z. Zhang, "Bottracer: Execution-based bot-like malware detection," in *Proc. of the 11th Information Security Conf., ISC*, 2008, pp. 97–113.
[17] Malware Domains, "DNS-BH Malware Domain Block List," 2013, http://www.malwaredomains.com/.
[18] McAfee, "McAfee Thread Reports," 2014, http://www.mcafee.com/apps/viewall/publications.aspx.
[19] S. Nagaraja, P. Mittal, C.-Y. Hong, M. Caesar, and N. Borisov, "Botgrep: Finding p2p bots with structured graph analysis." in *USENIX Security Symposium*, 2010, pp. 95–110.
[20] P. Porras, H. Saidi, and V. Yegneswaran, "An analysis of conficker logic and rendezvous points," *Computer Science Laboratory, SRI International, Tech. Rep*, 2009.
[21] L. A. Poyneer and J.-P. Veran, "Toward feasible and effective predictive wavefront control for adaptive optics," in *proc. SPIE*, vol. 7015, 2008, p. 70151E.
[22] K. Rieck, G. Schwenk, T. Limmer, T. Holz, and P. Laskov, "Botzilla: detecting the phoning home of malicious software," in *Proc. of the 2010 ACM Symposium on Applied Computing*, 2010, pp. 1978–1984.
[23] E. Stinson and J. C. Mitchell, "Characterizing bots remote control behavior," in *Proc. of the 4th Int'l Conf. on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2007, pp. 89–108.
[24] F. Tegeler, X. Fu, G. Vigna, and C. Kruegel, "Botfinder: finding bots in network traffic without deep packet inspection," in *Proc. of the 8th Int'l Conf. on Emerging networking experiments and technologies*. ACM, 2012, pp. 349–360.
[25] M. Vlachos, S. Y. Philip, and V. Castelli, "On periodicity detection and structural periodic similarity." in *SDM*, vol. 5, 2005, pp. 449–460.