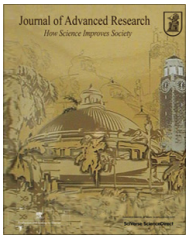
Journal of Advanced Research (2014) 5, 435–448

Cairo University

Journal of Advanced Research



ORIGINAL ARTICLE

An efficient method to detect periodic behavior in botnet traffic by analyzing control plane traffic

Basil AsSadhan [a,*], **José M.F. Moura** [b]

[a] *Department of Electrical Engineering, King Saud University, P.O. Box 800, Riyadh 11421, Saudi Arabia*
[b] *Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA*

ARTICLE INFO

Article history:
Received 20 September 2013
Received in revised form 17 November 2013
Accepted 21 November 2013
Available online 28 November 2013

Keywords:
Botnet detection
Control plane traffic
Discrete time series analysis
Walker large sample test
SLINGbot

ABSTRACT

Botnets are large networks of bots (compromised machines) that are under the control of a small number of bot masters. They pose a significant threat to Internet's communications and applications. A botnet relies on command and control (C2) communications channels traffic between its members for its attack execution. C2 traffic occurs prior to any attack; hence, the detection of botnet's C2 traffic enables the detection of members of the botnet before any real harm happens. We analyze C2 traffic and find that it exhibits a periodic behavior. This is due to the pre-programmed behavior of bots that check for updates to download them every T seconds. We exploit this periodic behavior to detect C2 traffic. The detection involves evaluating the periodogram of the monitored traffic. Then applying Walker's large sample test to the periodogram's maximum ordinate in order to determine if it is due to a periodic component or not. If the periodogram of the monitored traffic contains a periodic component, then it is highly likely that it is due to a bot's C2 traffic. The test looks only at aggregate *control plane* traffic behavior, which makes it more scalable than techniques that involve deep packet inspection (DPI) or tracking the communication flows of different hosts. We apply the test to two types of botnet, tinyP2P and IRC that are generated by SLINGbot. We verify the periodic behavior of their C2 traffic and compare it to the results we get on real traffic that is obtained from a secured enterprise network. We further study the characteristics of the test in the presence of injected HTTP background traffic and the effect of the duty cycle on the periodic behavior.

© 2013 Production and hosting by Elsevier B.V. on behalf of Cairo University.

Introduction

Botnets are large networks of bots (compromised machines) that are under the control of a small number of bot masters. In recent years, the threat posed by botnets toward Internet applications and communications has escalated. This is due to the fact that a bot master controls a large number of bots that ranges from hundreds of thousands to millions. This magnifies the impact of well-known network malicious activities such as scanning, E-mail spam and distributed denial-of-service (DDoS) attacks. Moreover, botnets increase the effectiveness of phishing, click fraud, identity theft, and espionage.

Due to the destructive capabilities of botnets, they have become a major threat to economy, information, and communication infrastructures. The Federal Bureau of Investigation (FBI) in the United States, in an initiative to detect bot masters

* Corresponding author. Tel.: +966 114676755; fax: +966 114676757.
E-mail address: bsadhan@ksu.edu.sa (B. AsSadhan).
Peer review under responsibility of Cairo University.

Production and hosting by Elsevier

2090-1232 © 2013 Production and hosting by Elsevier B.V. on behalf of Cairo University.
http://dx.doi.org/10.1016/j.jare.2013.11.005

and take them apart has identified over 1 million victim computers [1,2]. Many people have been indicted, pleaded guilty, or been sentenced for crimes related to botnet usage [1,2]. What increases the impact of the problem is that the majority of the owners of the compromised machines are not aware that their machines are a member of a botnet [1]. According to the April 2013 Symantec Internet Security Threat Report, Volume 18, 3.4 million distinct bot-infected computers were observed in 2012 [3]. According to the same report, botnets were responsible for about 69% of spam E-mail in 2012 [3]. The good news; there is a decrease in these numbers over the past years. For example, in 2009, there were 6.08 million distinct bot-infected computers and botnets were responsible for about 85% of spam E-mail [4]. Nevertheless, the numbers are still high; moreover, bot masters have begun linking mobile smart phones to form botnets of mobile devices to make monetary profits [3].

Botnets' traffic is different from the traffic of other types of malware in that it includes command and control (C2) communication channels traffic. A bot master relies on these channels to send commands to the members of its botnet to execute attack activities. In addition, a bot master relies on these channels to control botnet members to obtain the needed information and code to run their attacks. C2 communication channels traffic occurs before the execution of attack activities and can be considered as the intelligence communication between the different members of a botnet. This makes the detection of C2 communication channels traffic of interest as it means detecting bots before any targeted victim is attacked.

The detection of C2 traffic is difficult due to several reasons that was pointed out by AsSadhan et. al [5]. They include the following: (1) the low volume of C2 traffic; (2) C2's traffic is well behaved and does not violate any network protocol rules; (3) there may be only a few number of botnet members in the monitored network; and (4) the C2 traffic might be encrypted [5]. To tackle these difficulties, we look at one behavior that we, along with other researchers, observed in C2 traffic [5–7]. The behavior we observe is spatial-temporal correlation and similarities in the C2 communication traffic of the bots belonging to the same botnet.

In our work, we focus on temporal correlation in a single bot's traffic. We find that a bot's C2 traffic exhibits periodic behavior. This is due to the nature of the pre-programmed behavior of a bot, where in many variations of botnets each bot frequently contacts other bots every T seconds. This pre-programmed behavior is present in botnets with different structures and communication protocols and is done in order for bots to update their data, receive commands, and send keep-alive messages. We note that the periodic behavior is observed when looking at the traffic of the transport port number used by the bot for its C2 communication.

As a result, the detection of periodic behavior in a machine's traffic might be an indication that the machine is a member of a botnet. We exploit this observation in order to detect bots by detecting periodic behavior in the traffic of the network we monitor. To achieve this we present in this paper an efficient method to detect periodic behavior in botnet command and control traffic. The method is based on the evaluation of the traffic sequence's periodogram. A periodogram is used to view a periodic signal in the frequency domain to observe the peak located at the fundamental frequency of the signal. After the peak is located, we apply Walker's large sample test to decide whether or not the peak is significant enough compared to the rest of the periodogram's ordinates. In case the peak is significant, we declare that it is due to a periodic component with the frequency where the peak is located.

To increase the efficiency of the method further, we decompose enterprise LAN TCP traffic into control and data planes [8], and use the control plane traffic as a surrogate for the whole traffic (control and data planes combined). This is because data traffic generation is based on control traffic generation, which makes the behavior of the two traffic groups similar [8].

The rest of the paper is organized as follows, Section "Background and motivation: Detection of periodic behavior in botnet C2 communication channels traffic" reviews the command and control (C2) traffic of botnets and proposes how to detect botnets. Section "Approach: Discrete time series analysis of aggregate traffic" explains how we aggregate network traffic and decompose them into control and data planes traffic. Section "Methodology: Test network traffic for periodic behavior using periodograms" reviews periodograms and presents the Walker's large sample test. Section "Experimental setup: Evaluation and analysis" explains the experimental setup and presents our evaluation and analysis results of applying the test to several packets traces, and in Section "Conclusions" we give our conclusions.

Background and motivation: Detection of periodic behavior in botnet C2 communication channels traffic

Since a bot master controls a botnet via command and control (C2) communication channels. Our approach is to detect a botnet through the detection of its C2 communication channels traffic. This technique is effective as it detects bots before they engage in harmful malicious activities. This is because C2 traffic by itself is harmless, and its detection it will enable the detection of the bots that are transmitting/receiving it.

The C2 communication channels between bots and the C2 servers are based on either a pull or push mechanism [7]. Depending on the mechanism used, bots are pre-programmed to contact each other every T seconds to update bot's data, receive commands, and send *keep-alive* messages. This pattern of behavior is present in bots irrespective of the botnet's structure or the communication protocol being used between bots and the C2 server. This results in having a *periodic* behavior in the bot's traffic over a given transport port number. We note that in other botnet variants, C2 communication happens might occur in an aperiodic manner at arbitrary times. We briefly discuss this issue in Section "Experimental setup: Evaluation and analysis".

In our work, we exploit this periodic behavior to detect C2 communication traffic. In addition to our previous works [5,6], we are aware of a previous study, that exploits the periodic behavior of botnet C2 traffic to detect bots. In Gu et al. [7], the host's traffic autocorrelation function was computed in the time domain to examine whether the traffic has a periodic component or not. We, however, work in the frequency domain, as it involves less amount of computations, thus is faster in time. This is done thorough evaluating the periodogram [9] of the traffic and then applying Walker's large sample test [10] to the periodogram's maximum ordinate to detect periodic components.

The advantage of such technique is that it is based on an basic property shared by many botnet variants and is independent of the structure (e.g., centralized, P2P) and communication protocol (e.g., IRC, HTTP) used in the botnet. What is important is that it does not require a priori knowledge (e.g., signatures) of a certain botnet behavior, provided the C2 communication traffic exhibits periodic behavior. We note that network traffic in general can exhibit periodic behavior. Example of this would be an E-mail session that checks periodically for new messages. This can affect the accuracy of the detection of C2 traffic by introducing false positives. We address this issue in Section "Experimental setup: Evaluation and analysis".

Approach: Discrete time series analysis of aggregate traffic

To detect botnet C2 communication channels traffic, we apply discrete time series analysis to study the aggregate traffic behavior. To accomplish this, we have to extract a discrete time sequence from a packet trace. This is done by first aggregating packets originating from or destined to a given host, subnet, or network over an appropriate aggregation interval. Next, we extract a count-feature [8,11] from the packet header information to produce the discrete time sequence. The basic count feature is the packet count, which is the number of packets in that aggregation interval. Other examples of count-features include the byte count, which is the total number of bytes in all of the packets within the aggregation interval, and the different addresses count, which is the total number of distinct IP addresses in all of the packets within the aggregation interval. The selection of the aggregation interval is based on the packet rate of the traffic at the given host, subnet, or network. A higher packet rate implies using a smaller the aggregation interval. The objective is to avoid a discrete time sequence with low variance.

Having packet traces in the form of discrete time sequences enables applying statistical signal processing methods that are used in discrete time series analysis. Furthermore, monitoring aggregate traffic behavior requires keeping track of less details of the traffic when compared for example to tracking the communication flows of different hosts or examining the content of individual packets. We acknowledge that tracking less details of network traffic implies having less knowledge, which might lead to a lower accuracy in detection and higher false positive rates, but has several advantages. First, it consumes less computational processing, hence faster analysis in time; both contributing to higher scalability. We emphasize that with the larger growth in using high bit rate applications in the Internet, the demand for higher scalability has increased. Second, characterizing malicious activities traffic with less details can lead to a detection scheme that is more resistant to evasion. This is because the malicious attacker will have more restrictions to evade the characterized malicious activity without compromising the efficiency of the attack.

Discrete time series analysis of control plane traffic

The above analysis can be also applied to the *control plane* traffic packets [8]. Control plane traffic packets as AsSadhan et. al define [8] are the packets "that set, maintain, or tear down a connection". Data plane traffic packets are those packets that are involved in the transmission of the actual data. For TCP packets, we treat a packet as a control plane packet if it is one of these four types:

(1) SYN packet.
(2) Bare ACK packet, which is an acknowledgment packet with no payload.
(3) FIN packet.
(4) RST packet.

We apply discrete time series analysis to TCP *control* plane traffic due to its similar behavior to the data plane traffic as discussed previously [8]. The reason behind this similarity is that the generation of data traffic is based on the generation of the control traffic [8]. Therefore, analyzing the control plane traffic only we might suffices for analyzing the whole traffic (i.e., control and data planes). Such analysis reduces the amount of traffic to look at, which further implies fewer computations and faster analysis, both contributing to higher scalability. We note that since UDP is a connectionless unreliable protocol, the information in a packet's UDP header is not sufficient to decide whether to treat it as a control or data plane packet. Thus, unless we have access to the packet's application header and this header has sufficient information, it is not possible to decompose UDP packets into control and data planes packets as in the case of TCP.

Methodology: Test network traffic for periodic behavior using periodograms

We propose to detect periodic behavior in botnet C2 traffic by analyzing the Power Spectral Density (PSD) of the network traffic. The PSD can be estimated by taking the Fourier transform of the autocorrelation function. Alternatively, a PSD can be estimated using periodograms [9]. The periodogram of a time sequence (signal) provides its power at different frequencies. Periodograms have been used in other areas to detect periodic behavior like in biology [12] and geophysics [13]. It has also been used to analyze network traffic, see for example [14].

The periodogram is useful to identify frequency components that possess high power levels. Therefore, the periodogram of a periodic signal will have a high peak at the reciprocal of the fundamental period of the signal when compared to the mean of the periodogram. To evaluate the periodogram of a given traffic trace, we first extract a count-feature over a selected aggregation interval to produce a discrete time sequence $x[n]$. The periodogram $P_{xx}[k]$ of a discrete time sequence $x[n]$ is the square magnitude of the Discrete Fourier Transform (DFT) of the signal evaluated by

$$P_{xx}[k] = \frac{1}{N}|X[k]|^2,$$

where

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp\left(\frac{-j2\pi kn}{N}\right)$$

is the N-point DFT.

Fig. 1 illustrates how periodograms can be used to detect periodic behavior. The top-left plot shows a periodic train of rectangular pulses with levels 0 and 1, a period of 0.1 s, a duty cycle of 20%, and a duration of 1 s. The one sided

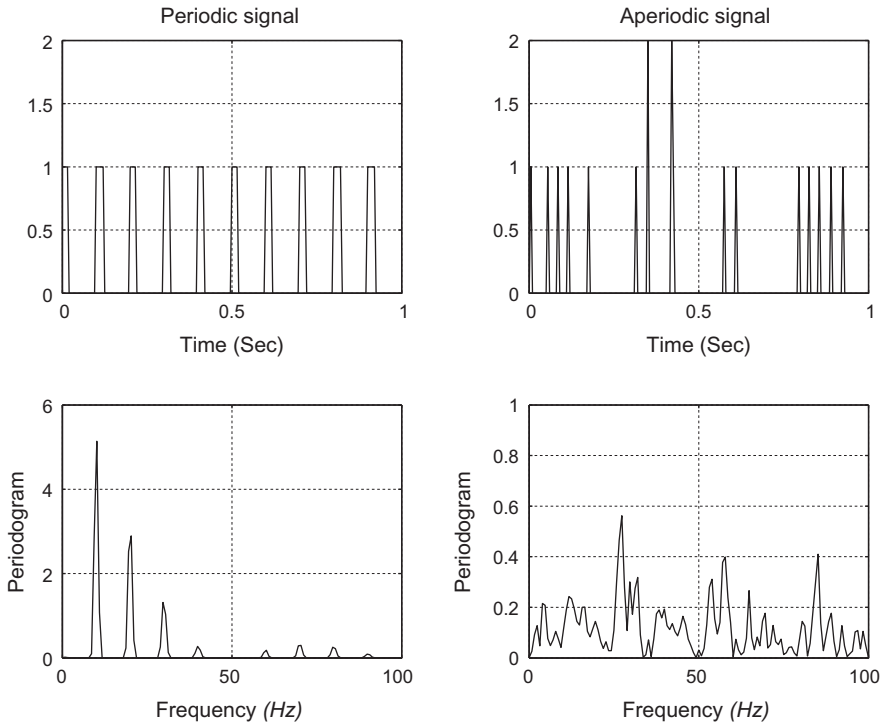


Fig. 1 Left plots show a periodic train of rectangular pulses with a period of 0.1 s, and its one sided periodogram that consists of a large peak at the fundamental frequency and smaller peaks at the harmonics. Right plots show an aperiodic signal, and its one sided periodogram that consists of several small peaks.

periodogram of this periodic signal after subtracting its mean is shown in the figure's bottom-left plot. The periodogram consists of a large peak at 10 *Hz*, smaller peaks at multiples of 10 *Hz*, which represent the harmonic components, and almost zero elsewhere. The figure's top-right plot shows an aperiodic Poisson random signal with a variance of 0.16, and a duration of 1 s. The selection of the variance was made in order to have it equal to the variance of the square wave signal. The one sided periodogram of the aperiodic signal after subtracting its mean is shown in the figure's bottom-right plot. It consists of several peaks; none of them has a significantly large value when compared to the mean of the periodogram.

The periodogram we described above is referred to as the standard periodogram. We note that we choose not to use the Welch's method of averaged periodogram [9,15]. This is because our work is interested in the detection and estimation of a single periodic component, which is better achieved using the standard periodogram as discussed in So et al. [16]. The averaged periodogram is used by others to represent the spectral density of the traffic [14] and to detect a periodic component with non-stationary phase [16]. We use however the modified periodogram [9]. The modification is done by multiplying the traffic sequence by a Hamming window in order to reduce the level of the side-lobes. We note that windowing comes at the cost of reducing the sharpness of the peak.

Detecting the significance of the periodogram's peak

From Fig. 1's bottom plots, we can see that the periodogram of any signal whether it is periodic or aperiodic will always have a peak. Thus, we need to have the ability to decide whether the peak is significant enough when compared to the rest of the periodogram's ordinates to declare that the sequence contains a periodic component or not. We use binary hypothesis testing [17,18] to achieve this.

Before setting up the two hypotheses test, we state a basic assumption we adopt; the count-feature sequence extracted from the network traffic communication of a given host on a given port has a Poisson distribution. We acknowledge that Poisson statistics may or may not be an accurate model for network traffic at a given host on a given port. We use it to simplify the analysis. Other exponential family models will affect the threshold selection, but not the test itself.

The Poisson distribution is a good approximation to the binomial distribution when the binomial parameter n is large and the binomial parameter p is small[1] [19]. A binomial random variable with a large n and its Poisson approximation (when p is small) can be, based on the central limit theorem, approximated by a Gaussian random variable [19]. Therefore, the count-feature sequence extracted from the packet trace after subtracting its mean and normalizing it by its standard deviation, can be treated as a standard Gaussian distribution (i.e., $N(0, 1)$).

We now set up the null hypothesis H_0; the count-feature sequence $x[n]$ is Gaussian, against the alternative hypothesis H_1; $x[n]$ has a periodic component at some unspecified frequency plus Gaussian noise. Under H_0, it can be shown that the ordinates $P_{xx}[k_0]$ of the periodogram of $x[n]$ are independent

[1] A binomial random variable is defined as the sum of n independent identically distributed (i.i.d.) Bernoulli random variables with probability p.

identically distributed (i.i.d.) [10]. Each $P_{xx}[k_0]$ has a distribution that is proportional to a chi-square distribution with two degrees of freedom [10]. Specifically,

$$P_{xx}[k_0]/\sigma_x^2 = \chi_2^2.$$

Since a chi-square distribution with two degrees of freedom is equivalent to an exponential distribution with mean 2, it follows that the probability density function of $P_{xx}[k_0]/\sigma_x^2$ is

$$f(x) = \frac{1}{2}\exp(-x/2), \quad 0 \le x < \infty$$

Therefore, for $z \geqslant 0$,

$$\begin{aligned}\Pr[P_{xx}[k_0]/\sigma_x^2 \le z] &= \int_0^z \frac{1}{2}\exp(-x/2)dx \\ &= 1 - \exp(-z/2). \end{aligned} \quad (1)$$

Since we are interested in the periodogram's ordinate that has the maximum value, we define the ratio test statistics,

$$\gamma_x = \frac{\max_{0\le k\le m-1}(P_{xx}[k])}{\sigma_x^2}. \quad (2)$$

Since under H_0, the periodogram ordinates $P_{xx}[k_0]$ are i.i.d., then it follows that, for $z \geqslant 0$,

$$\begin{aligned}\Pr[\gamma_x > z] &= 1 - \Pr[\gamma_x \le z] = 1 - \Pr[(P_{xx}[k_0]/\sigma_x^2) \\ &\le z, \text{all } k_0] = 1 - (1 - \exp(-z/2))^m, \end{aligned} \quad (3)$$

where m is the number of ordinates at the positive frequencies of the periodogram.

Eqs. (1)–(3) assume that the variance σ_x^2 is known a priori. However, in practice, it is typically unknown, and an estimate is used. The variance σ_x^2 can be estimated directly from the time sequence $x[n]$ using the sample variance. But since $x[n]$ might not be available, it is better to estimate σ_x^2 directly from $P_{xx}[k]$. The estimate of σ_x^2 according to Priestley [10] can be evaluated by

$$\widehat{\sigma}_x^2 = \frac{1}{2m}\sum_{k=0}^{m-1} P_{xx}[k]$$

The quantity $\widehat{\sigma}_x^2$ is an unbiased estimate of σ_x^2, and we will use it in place of σ_x^2 in (2) to define the sample ratio test statistic,

$$g_x^* = \frac{\max_{0\le k\le m-1}(P_{xx}[k])}{\frac{1}{2m}\sum_{k=0}^{m-1} P_{xx}[k]}. \quad (4)$$

When m is large, $\widehat{\sigma}_x^2$ will be a good approximation to σ_x^2; thus, we can treat the denominator of (4) as σ_x^2. Then, g_x^* will have the same distribution as γ_x, and asymptotically under H_0 we have, for $z \geqslant 0$,

$$\Pr[g_x^* > z] \sim 1 - (1 - \exp(-z/2))^m. \quad (5)$$

The asymptotic distribution of g_x^* is the basis of Walker's large sample test for max $(P_{xx}[k])$, [10].

Under the alternative hypothesis H_1, where the signal is periodic, the sample ratio test statistic g_x^* will be large. This enables us to use a one sided test and select the critical region $g_x^* > z_\alpha$, where z_α is selected so the right hand side of (5) is equal to α, which represents the false positive probability of the test. If the calculated value of g_x^* from the sample data is less than z_α, we then accept H_0, and conclude that $x[n]$ does not have any periodic component. If g_x^* is larger than z_α, we then reject H_0, with a false positive probability of α and conclude that $x[n]$ has a periodic component. The value of α is selected based on how small we would like the false positive probability to be.

We note that Fisher has derived an exact test for max $(P_{xx}[k])$ [10,12,20]. However, we use Walker's test for the following reasons: first, Fisher's test involves using combinatorial coefficients, which are limited in their accuracy as the number of sample points gets large; hence we lose the exactness of the test. Second, even if the number of sample points is not large, evaluating z_α in the Fisher test to set the critical region to α is not straight forward. Instead, we need for each measured g_x^* to evaluate the probability that it would be greater than this value, and then check if the probability is smaller than α or not. Third, usually, we are not short of network traffic to get a good estimate $\widehat{\sigma}_x^2$ to use in the denominator of (4).

In Walker's test described above, when the peak of the periodogram, max $(P_{xx}[k])$, is significant, we can only declare that the count-feature sequence has a periodic component at some frequency f. The question would be can we conclude that the count-feature sequence has a periodic component with the frequency where max $(P_{xx}[k])$ is located. Hartley has answered the question and showed that the probability that the sequence has a periodic component at some other frequency f is less than α, the probability of false positive, [10]. Therefore, when the peak of the periodogram, max $(P_{xx}[k])$, is significant, we can conclude that the count-feature sequence contains a periodic component with the frequency where max $(P_{xx}[k])$ is located.

Experimental setup: Evaluation and analysis

We utilize SLINGbot [21] (**S**ystem for **L**ive **I**nvestigation of **N**ext **G**eneration **bot**s) to generate examples of botnet C2 traffic. The traffic includes downloading bot software, connecting to bot C2 servers, and receiving botnet commands. SLINGbot uses a C2 feature space that consists of five separate dimensions for the functionality of botnets. The five dimensions are the following: topology, rallying mechanism, communication protocol, control mechanism, and command authentication mechanism. We use SLINGbot to set three variants of botnets, two TinyP2P[2] and one IRC, which we discuss further in this section.

Periodic behavior in packet and address counts of botnet C2 traffic

We use SLINGbot to set five bots and one bot master in a mesh topology; the bots use TinyP2P as its protocol of communication on port number 11375. Each bot is pre-programmed to update its data every 3 s by contacting other bots. The experiment was run for approximately 20 s. Since we notice that the C2 traffic of each of these six bots is similar, we will only discuss the traffic of one of them. We use an aggregation interval of 100 ms to extract the packet and address count sequences from the bot's traffic.

Fig. 2's top plots show the packet and address count sequences of the C2 communication traffic of a TinyP2P bot. The periodic behavior is apparent in both count sequences. At the start of each period, each bot contacts other bot to check whether there are any updates, and if so it downloads them; after that, it becomes silent until the next period starts.

[2] A TinyP2P botnet is a botnet that uses TinyP2P as its communication protocol [21].

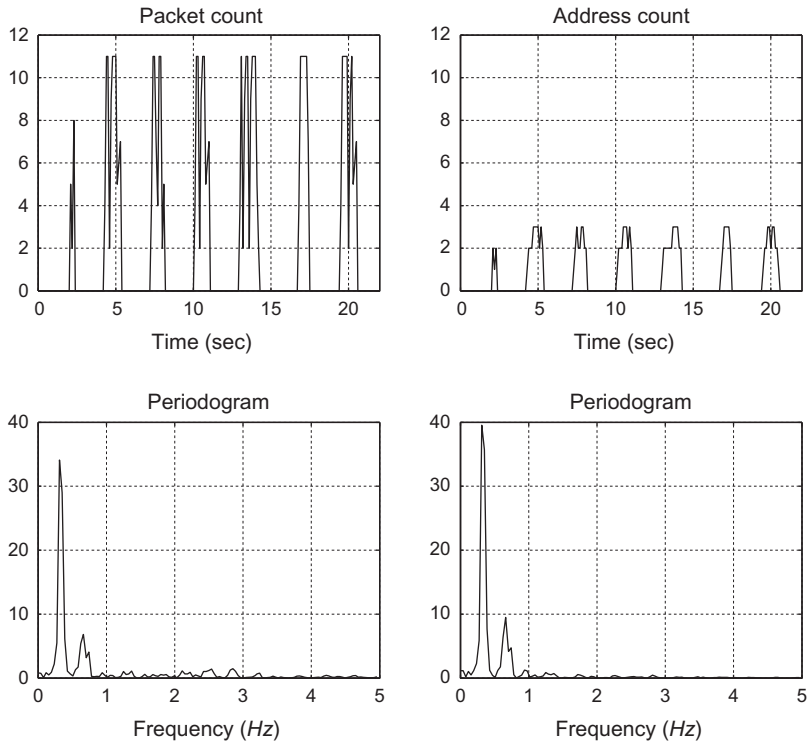


Fig. 2 Left plots show the packet count for the C2 communication traffic of TinyP2P bot and its one sided periodogram. Right plots show the address count for the same traffic and its one sided periodogram. The aggregation interval for the packet and address counts is 100 ms.

The bottom plots in the same figure show the modified periodograms of these count sequences after subtracting their means and normalizing them by their standard deviations.

In both plots, there is a single major peak at 313 mHz. This peak corresponds to a period of 3.2 s, which agrees with the 3-second pre-programmed period of the bot. We also notice that the peak of the periodogram of the address count sequence has a higher value than the one of the packet count sequence. This is due to the number of distinct addresses in the traffic flow, which has fewer fluctuations when compared to the number of packets.

We select the false positive probability, α, to be equal to 0.1%. This value is selected since, in general, it is desired to have a low false positive rate, in particular, in network anomaly detection systems. We equate the right hand side of (5) to α, to get a threshold, $z_{0.1\%}$, of 23.5.[3] We test the significance of the peak, max ($P_{xx}[k]$), of the two periodograms by evaluating the sample ratio test statistic g_x^* in (4) for both periodograms. The ratio value is found to be 61.7 and 70.7 for the periodogram of the packet and address count sequences, respectively. Since the two values of g_x^* are larger than $z_{0.1\%}$, we reject the null hypothesis and conclude that both sequences contain a periodic component with a frequency of 313 mHz.

The effect of duty cycle on periodic behavior

We use SLINGbot to set another TinyP2P botnet that consists of five bots and one bot master. Each of the five bots is pre-programmed to update its data with a different period; the periods that the bots use are 3, 4, 5, 6, and 7 s. The experiment was run for approximately 35 s. We use an aggregation interval of 100 ms to extract the packet count of each bot's traffic.

Fig. 3's top plots show the packet count sequences of two TinyP2P bot C2 traffic traces. The periods of the traffic for the first bot (top-left) and the second bot (top-right) are 4 and 6 s, respectively. As can be seen, the periodic behavior is apparent in both plots. Fig. 3's bottom plots show the modified periodograms of the packet sequences after subtracting their means and normalizing them by their standard deviations. The periodogram of the first bot (bottom-left) consists of a large peak at 273 mHz. This peak corresponds to a period of 3.7 s, which agrees with the 4-second pre-programmed period of the bot. The periodogram of the first bot also consists of smaller peaks at the multiples of the frequency. The smaller peaks are the harmonic components as discussed in Section "Methodology: Test network traffic for periodic behavior using periodograms". The same observations apply to the periodogram of the second bot (bottom-right), except that the peak is located at 176 mHz, which corresponds to a period of 5.7 s. Similar to what we get in the first TinyP2P botnets, this period agrees with the 6-second pre-programmed period of the bot. We test the significance of the peak, max ($P_{xx}[k]$), of each of the two periodograms by evaluating the sample ratio test statistic g_x^* in (4) for each periodogram.

We select the false positive probability, α, to be equal to 0.1%. We equate the right hand side of (5) to α, to get a threshold, $z_{0.1\%}$, of 24.9.[4] We obtain the values of g_x^* for each of the

[3] The number of ordinates at the positive frequencies of both periodograms, m, used in evaluating, $z_{0.1\%}$ is 128.

[4] The number of ordinates at the positive frequencies of both periodograms, m, used in evaluating, $z_{0.1\%}$ is 256.

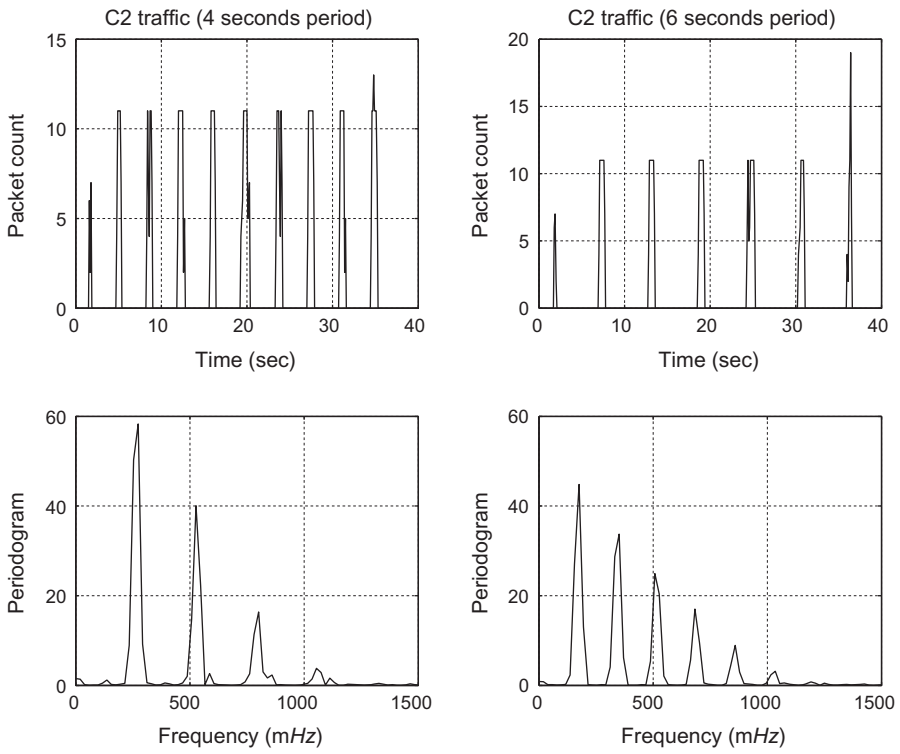


Fig. 3 The top plots show the packet count sequences for the C2 communication channels traffic of two TinyP2P bots with two different periods. The bottom plots show the one sided periodogram of each sequence. The aggregation interval for the packet count is 100 ms.

two periodograms to be 101.4 and 77.0; both are larger than $z_{0.1\%}$. Therefore, we reject the null hypothesis and conclude that both sequences have a periodic component. We note that the two reported values of g_x^* in the packet count sequences of the second TinyP2P botnet C2 traffic traces (101.4 and 77.0) are higher than the one reported for the packet count of the first TinyP2P botnet C2 traffic traces (61.7). This is because the second botnet was run for a longer time (35 s), while the other one was run for a shorter time (20 s). This results in having a larger number of periods, which increases the traffic's periodogram peak value.

Duty cycle

For a periodic train of pulses, the *duty cycle* is the ratio of the pulse duration divided by the period's length. We note that the active time duration of the packet count sequences shown in Fig. 3's top plots is not affected by the period's length in the two bots. Since the first bot's period is smaller than the second bot's period, the first bot's traffic duty cycle is higher than the second bot's traffic duty cycle. This causes the periodogram of the first bot to have a value of g_x^* (101.4) that is higher than the value of g_x^* for the second bot (77.0) because the ratio of the periodogram's peak to the harmonic components is larger. This is due to the following fact, which is illustrated in Fig. 4. The periodogram of a train of rectangular pulses is a sampled squared sinc function. For trains of rectangular pulses with a larger duty cycle, the ratio between the main peak and the harmonic components at the Periodogram is also larger. Hence, for trains of rectangular pulses with larger duty cycle, the main peak is more significant. Fig. 4's top-left plot shows a periodic train of rectangular pulses with a period of 10 s and a duty cycle of 25%. The top-right plot shows a similar periodic train of rectangular pulses, but with a duty cycle of 50%. The bottom plots show the periodograms of these two periodic signals after subtracting their means and normalizing them by their standard deviations. It can be seen that the periodic signal that has the higher duty cycle has a higher ratio between the periodogram's main peak and the harmonic components.

Periodic behavior in control plane traffic

We use SLINGbot to set 20 bots and one bot master in a star topology that uses IRC (Internet Relay Chat) as its protocol of communication on port number 6667. The bot master is pre-programmed to broadcast its updates to the 20 bots every 5 s. The experiment was run for approximately 40 s. Since we notice that the C2 traffic in each of these 21 bots is similar, we will only analyze the traffic of one of them.

Fig. 5's top plots show the packet and address count sequences for the C2 communication traffic of an IRC bot. As in the TinyP2P botnets, the periodic behavior is apparent in both count sequences. The bottom plots in the same figure show the modified periodograms of these sequences after subtracting their means and normalizing them by their standard deviations. In both plots, there is a single major peak at 195 mHz. This peak corresponds to a period of 5.1 s, which agrees with the 5-second pre-programmed period of the bot. As in the TinyP2P bot, the peak, max ($P_{xx}[k]$), of the periodogram of the address count has a higher value than the one for the periodogram of the packet count.

The two values of g_x^* for the packet and address count sequences of the IRC botnet C2 traffic are 40.8 and 46.2,

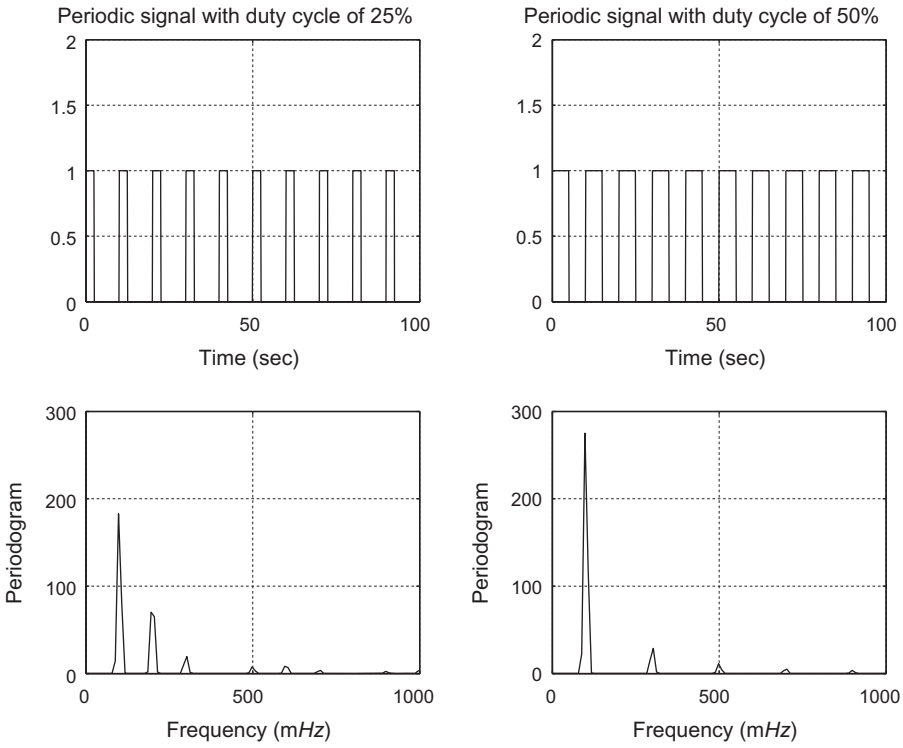


Fig. 4 Left plots show a periodic train of rectangular pulses with a period of 10 s and a duty cycle of 25%, and its one sided periodogram. Right plots show a periodic train of rectangular pulses with the same period and a duty cycle of 50%, and its one sided periodogram. The ratio between the periodogram's main peak and the harmonic components is higher in the periodic signal that has the higher duty cycle.

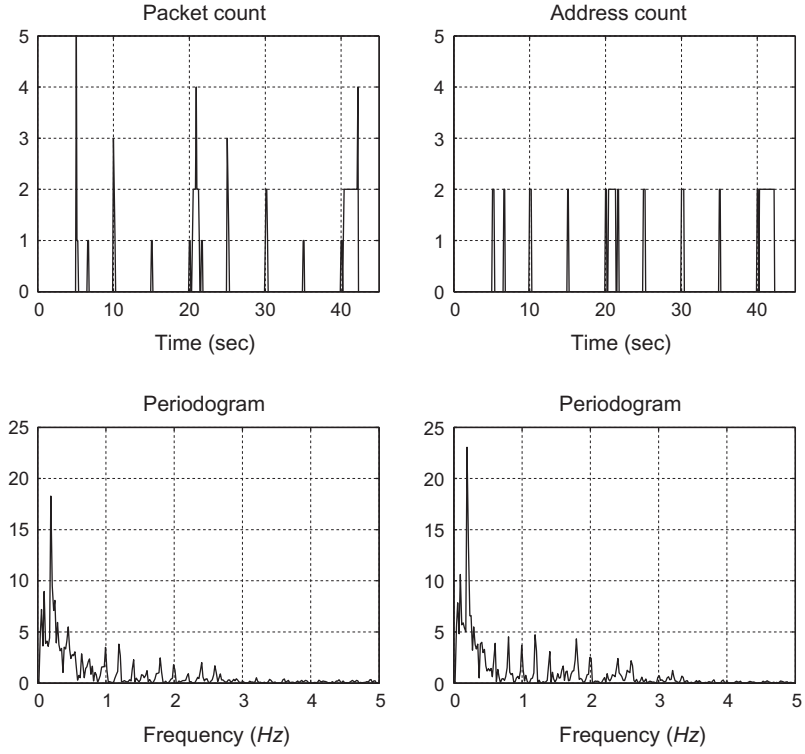


Fig. 5 Left plots show the packet count for the C2 communication traffic of an IRC bot and its one sided periodogram. Right plots show the address count for the same traffic and its one sided periodogram. The aggregation interval for the packet and address counts is 100 ms.

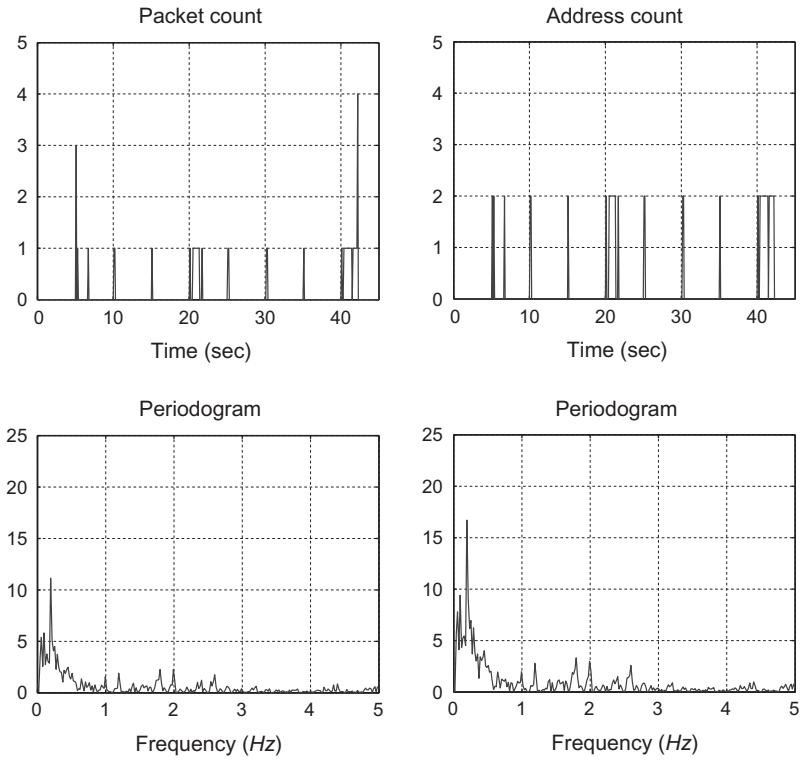


Fig. 6 Left plots show the packet count for the control plane of the C2 communication traffic of an IRC bot and its one sided periodogram. Right plots show the address count for the same traffic and its one sided periodogram. The aggregation interval for the packet and address counts is 100 ms.

respectively. These two values are larger than $z_{0.1\%} = 24.9$.[5] Thus, we can conclude that both sequences contain a periodic component with a frequency of 195 mHz. However, the two values are not as large as the ones in the sequences of the TinyP2P botnet traffic. This makes it easier for the bot master to hide the periodic behavior of the C2 traffic by adding background traffic on the same port to bury its C2 traffic, which we address in Section ''Experimental setup: Evaluation and analysis''. The low value of g_x^* is attributed to the low duty cycle of the count sequence, which resulted in higher harmonics in the frequency domain when compared to the periodogram's peak.

Control plane traffic

We study at the *control plane traffic* of the IRC bot C2 traffic. The plots in Fig. 6 show the packet and address count sequences of the IRC bot C2 control plane traffic and their modified periodograms. The plots look very similar to the ones in Fig. 5, where the packet and address count sequences are extracted from the control and data planes traffic trace.

Although the control plane packets in the IRC bot traffic constitutes 47% of the total number of packets in the control and data planes traffic, their volume (in bytes) is only 2.3% of the total size of the control and data planes traffic packets. Monitoring this much smaller traffic reduces significantly the processing time and effort. The impact of only monitoring the control plane traffic is reducing on the value of g_x^* from 40.8 to 35.7 in the case of the packet count sequence and from 46.2 to 34.9. Both values are still clearly larger than $z_{0.1\%} = 24.9$.

Periodic behavior in the presence of background traffic

A bot master can attempt to evade the detection of its botnet members by hiding the periodic behavior of the C2 traffic. This can be done in two ways; first, the bot master can carry the C2 communication channel session over a common port number used by other Internet applications (e.g., HTTP on port 80). This will decrease the percentage of C2 traffic's volume over this port; hence, its periodic behavior may not be noticeable. Alternatively, the bot master can program the traffic over a pre-determined randomized sequence of port numbers instead of having the C2 traffic between different bots over a single port number. Such evasion schemes might prevent the detection of the periodic behavior of a bot's C2 traffic by relying on monitoring the traffic of a host *per transport port numbers*.

In this section, we check whether we can continue to detect the periodic behavior of the C2 traffic of a *given bot* in the presence of background traffic. We use the packet traces collected by the Lawrence Berkeley National Laboratory/ International Computer Science Institute (LBNL/ICSI) Enterprise Tracing Project [22,23] as our background traffic. The packet traces were collected from two internal network locations at LBNL. LBNL is a research institute in the USA with a medium-sized enterprise network. Only packet header information is released to the public.

[5] The number of ordinates at the positive frequencies of both periodograms, m, used in evaluating , $z_{0.1\%}$ is 256.

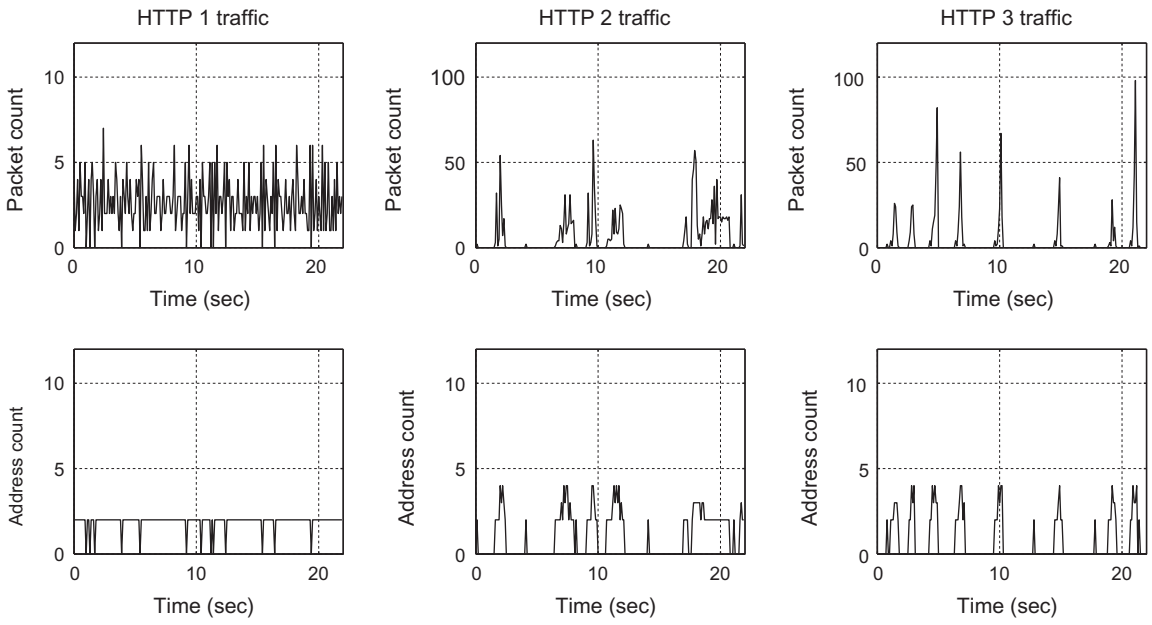


Fig. 7 The packet (top) and address (bottom) count sequences for three HTTP traffic traces in the LBNL/ICSI data set.

We use the HTTP traffic from the LBNL/ICSI packet trace as our background traffic, $y[n]$. We then add it to the C2 traffic of a TinyP2P bot, $x[n]$. The total traffic becomes

$$z[n] = x[n] + y[n],$$

and the resulting ratio g_z^* becomes noisy.[6] We use the Signal-to-Noise Ratio (SNR) to quantify the level of background traffic added. The SNR is the ratio of the power of the count-feature sequence of the botnet C2 traffic after subtracting its mean to the power of the count-feature sequence of the background traffic after subtracting its mean. In other words, the SNR is the ratio of the variances of the two sequences. The following formula is used to evaluate the SNR in dB:

$$\mathrm{SNR} = 10\log\frac{\hat{\sigma}_x^2}{\hat{\sigma}_y^2} = 10\log\frac{\frac{1}{N}\sum_{n=0}^{N-1}(x[n]-\hat{\mu}_x)^2}{\frac{1}{N}\sum_{n=0}^{N-1}(y[n]-\hat{\mu}_y)^2},$$

where N is the number of sample points of $x[n]$ and $y[n]$.

Fig. 7's plots show the packet (top) and address (bottom) count sequences for three HTTP traffic traces at port 80 in the LBNL/ICSI data set. The three traces were collected on October, 4, 2004. The IP addresses of the hosts that generated the traffic along with the start times (in PDT) of the traffic traces are as follows: HTTP 1 traffic: (131.243.86.124, 1:46:12 pm), HTTP 2 traffic: (131.243.125.239, 1:25:05), and HTTP 3 traffic (131.243.219.252, 2:53:02 pm). We note that the traffic of the three hosts get more bursty as we go to the right of the figure, specifically when we look at the packet count sequences. We add the traffic of each of these three hosts to the C2 traffic of the TinyP2P bot shown in Fig. 2 to observe the effect of this on the sample ratio test statistic g_z^*.

Fig. 8's plots show the packet and address count sequences for the TinyP2P bot C2 traffic after adding the HTTP 1 traffic trace to it. The periodic behavior is still apparent in both plots and the period is still found to be 3.2 s. The periodic behavior can also be noticed from the modified periodograms that are plotted in the bottom plots in the same figure after subtracting their means and normalizing them by their standard deviations, where a distinguished peak is still located at the frequency of the sequences at 313 mHz. However, the two values of g_z^* for the packet and address count sequences of the total traffic (botnet C2 and background traffic) are less than the ones we had for g_x^* in the absence of background traffic. In the case of the packet count, the value has decreased from 61.7 to 54.1, and in the case of the address count, it decreased from 70.7 to 57.7. Both values are still higher than $z_{0.1\%}$ (23.5), hence, the test declares both sequences to have a periodic component. The two values of g_z^* are still high because the background traffic had a low variance of, which resulted in a high SNR of 8.3 dB and 7.3 dB in the packet and address count sequences, respectively.

The case is different when we add either the HTTP 2 or HTTP 3 traffic traces to the TinyP2P bot C2 traffic as shown in Figs. 9 and 10. The periodic behavior is no longer apparent in the top plots of both the packet and address count sequences. This is due to the bursty nature of the HTTP 2 and HTTP 3 traffic traces that resulted in a low SNR. When adding HTTP 2 traffic, the SNR values are −9.2 dB and −1.3 dB for the packet count sequence and B the address count sequences, respectively. When adding HTTP 3 traffic, the SNR values are−9.8 dB and −1.3 dB for the packet count sequence and address count sequences, respectively. We make a note of the large difference in the SNR values between the packet and address count sequences. The very low SNR values in the packet count sequences cause the values of g_z^* to reach 21.4 and 14.6 when we add the HTTP 2 traffic and HTTP 3 traffic, respectively. Both values are lower than $z_{0.1\%}$ (23.5), hence, the test declares that the packet count sequences do not have a periodic component.

Contrary to the SNR values in the packet count sequences, they are not that low in the address count sequences. They only cause the values of g_z^* to reach 46.6 and 49.5 when we add the HTTP 2 traffic and HTTP 3 traffic, respectively. Both values

[6] We treat the background traffic here as noise traffic.

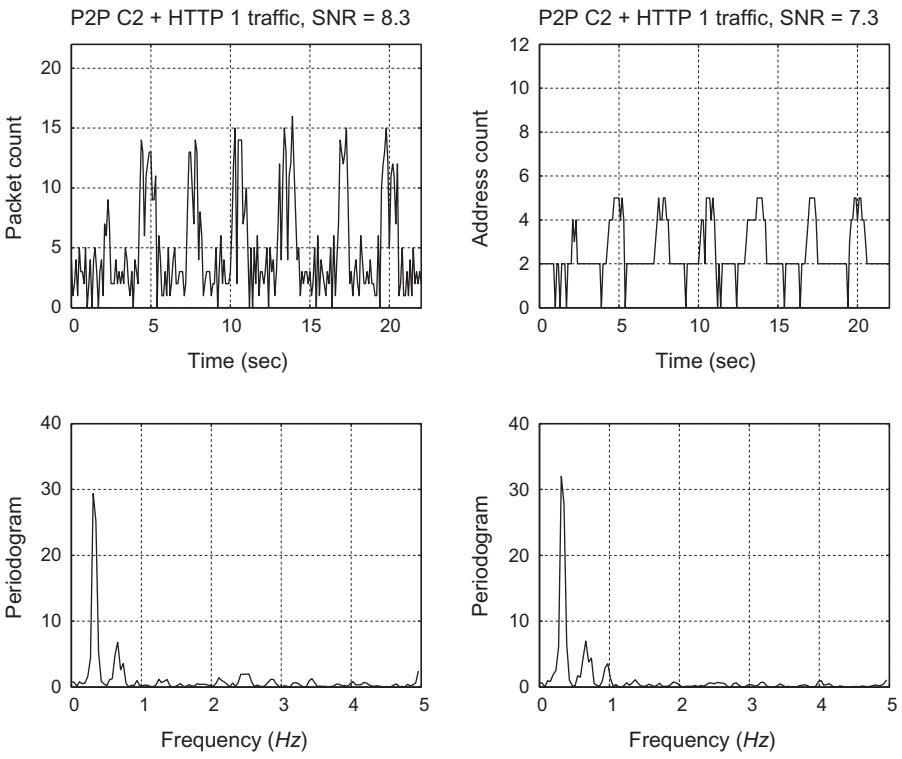


Fig. 8 The packet and address count sequences with their one sided periodograms of the C2 communication traffic of the TinyP2P bot shown in Fig. 2 with the HTTP 1 traffic shown in Fig. 7 added to it. The aggregation interval for the packet and address counts is 100 ms.

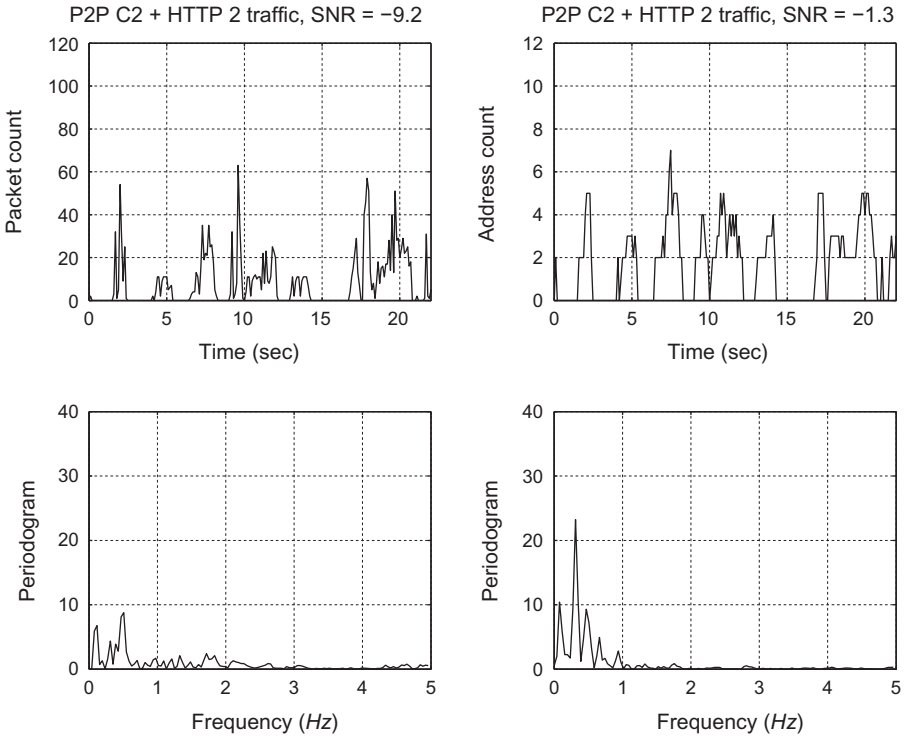


Fig. 9 The packet and address count sequences with their one sided periodograms of the C2 communication traffic of the TinyP2P bot shown in Fig. 2 with the HTTP 2 traffic shown in Fig. 7 added to it. The aggregation interval for the packet and address counts is 100 ms.

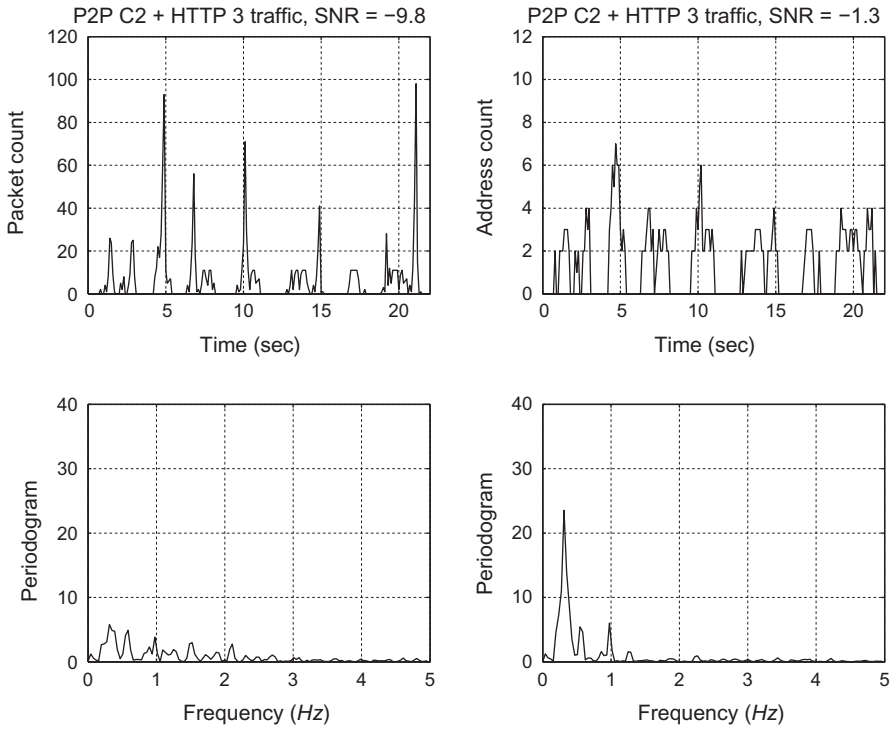


Fig. 10 The packet and address count sequences with their one sided periodograms of the C2 communication traffic of the TinyP2P bot shown in Fig. 2 with the HTTP 3 traffic shown in Fig. 7 added to it. The aggregation interval for the packet and address counts is 100 ms.

are still significantly higher than $z_{0.1\%}$ (23.5), hence, the test declares that the address count sequences have a periodic component. The periodic behavior can also be seen from the modified periodograms that are plotted in the bottom-right plots in Figs. 9 and 10 after subtracting their means and normalizing them by their standard deviations, where a distinguished peak is located at 313 mHz. We explain the reason behind the different observations between the packet and address count sequences. This is because of the fact that the number of distinct addresses (i.e., hosts) that a given host communicates with during a given aggregation interval is much smaller than the number of packets it would send/receive from these hosts. This makes address count sequences more robust to evasion schemes than packet count sequences.

The previous plots show the effect of adding the traffic of a single HTTP connection to a bot's C2 traffic. To examine the effect of adding the traffic of multiple HTTP connections to a bot's traffic, we add all of the three HTTP traffic traces to the C2 traffic trace. The results are shown Fig. 11's plots. The result we find in the packet count sequences is similar to the one we get after adding either the HTTP 2 or HTTP 3 traffic traces individually; i.e., the traffic trace no longer exhibits periodic behavior due to the low SNR. In the case of the address count sequences, the SNR is −4.3 dB, which is (in absolute value) half of the SNR value (−1.3 dB) when either the HTTP 2 or HTTP 3 traffic traces were added. The SNR value of −4.3 dB causes the value of g_z^* to reach 35.8, which is still higher than $z_{0.1\%}$ (23.5); hence, the test declares that the sequence has a periodic component. The periodic behavior can be seen from the modified periodograms that are plotted in the bottom-right plot in Fig. 11 after subtracting their means and normalizing them by their standard deviations, where a distinguished peak is located at 313 mHz.

To summarize, the detection of periodic behavior in botnet C2 traffic in the presence of background traffic depends on the count-feature sequence we test. In the case of packet count sequences, the test detects the periodic behavior down to a certain SNR level; below that level, it fails to detect the periodic behavior. We note that the results of adding background traffic to packet count sequences are similar to the results of AsSadhan et. al [6] when random noise traffic was injected. In the case of address count sequences, the test is much more robust to the background traffic and succeeds in detecting the periodic behavior. This is because of the fact that the number of distinct addresses (i.e., hosts) that a given host communicates with during a given aggregation interval is much smaller than the number of packets it would send/receive from these hosts. This illustrates that address count sequences are more robust to background traffic than packet count sequences.

Method's limitations

We address some of the limitations of basing the detection of botnet C2 communication traffic on the detection of its periodic behavior. The bot master can attempt to hide the periodic behavior of its bots by uniformly randomizing the period within a certain small range. This can be modeled by a random phase. The detectability of the periodic behavior here will depend on how large the random phase is and on the period's length and the duty cycle. In case the bot master uses a larger range, such that the signal is no longer periodic, it will succeed in evading our test. However, such evasion scheme will limit the effectiveness of the exchange of C2 channel traffic between different bots. This will result in not having C2 updates at pre-determined times, which might disturb the effectiveness

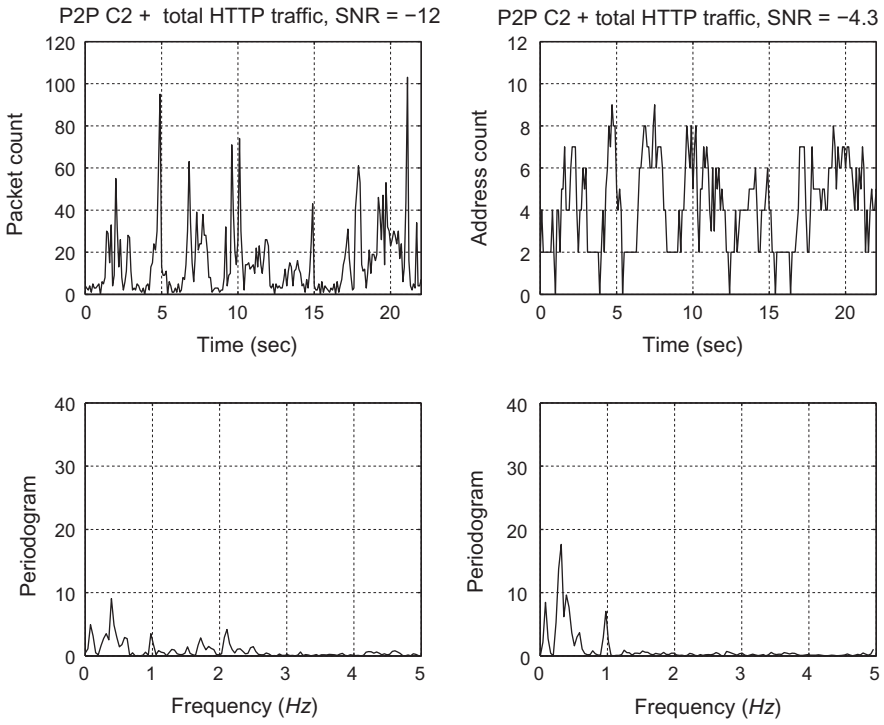


Fig. 11 The packet and address count sequences with their one sided periodograms of the C2 communication traffic of the TinyP2P bot shown in Fig. 2 with all of the three HTTP traffic trace shown in Fig. 7 added to it. The aggregation interval for the packet and address counts is 100 ms.

of the attack carried out by the botnet. In addition, there are certain applications that have a periodic nature. The traffic of these applications can introduce false positives, however, they can be easily white listed once we are aware of them. Alternatively, the output traffic of the test can be tested further using a more complex method to determine whether the observed periodic behavior is due to botnet C2 traffic or not. We note that such complex method might not scale up if applied directly to the original traffic trace. Therefore, our method can serve as a scalable first stage.

Conclusions

We propose a method that detects periodic behavior in network traffic. The method starts with the evaluation if the periodogram of the traffic sequence and then it locates its peak. After that, it uses Walker's large sample test to determine whether the periodogram's peak is significant when compared to the rest of the periodogram's ordinates or not. If it is determined to be significant, it declares the presence of a periodic component whose frequency is where the peak is located.

We use this method to detect botnet command and control (C2) channels traffic by detecting their periodic behavior. Periodic behavior arises in botnet C2 traffic since in many botnet variants bots are pre-programmed to check for and download updates every T seconds. We generate two variants of botnet C2 communication traffic using SLINGbot, TinyP2P and IRC. We apply Walker's large sample test to the C2 traffic and show that the traffic in both botnets exhibits periodic behavior in Figs. 2 and 3 and 5 and 6. The periodic behavior was also detected when analyzing the control plane traffic only. Since the volume of control plane traffic is much smaller, monitoring it will considerably reduce the processing time and effort.

We examine the effect of the duty cycle, length of observed traffic, and period length of the C2 traffic on the test performance. We show that the test's performance increases with the increase in the duty cycle and/or the length of the observed traffic, and decreases with the decrease in the period length. We study cases where the bot master attempts to evade the detection of the periodic behavior of C2 traffic by mixing it with HTTP background traffic. Our results show that, depending on the count-feature that is being tested, periodic behavior can still be detected. When testing the packet count sequence, periodic behavior is detected up to a certain packet volume level; above that level, periodic behavior is no longer detected. When testing the address count sequence, the test is much more robust and succeeds in detecting the periodic behavior. This is due to the fact that the number of hosts that a given host communicates with during a given aggregation interval is much smaller than the number of packets it sends/receives from these hosts.

Conflict of interest

The authors have declared no conflict of interest.

Acknowledgments

Basil AsSadhan extends his appreciation to the Deanship of Scientific Research at King Saud University for funding this

work through Research Project No. NFG2-02-33. We gratefully acknowledge the discussions of Dr. David Lapsley, Dr. W. Timothy Strayer, Dr. Alden Jackson, and Ms. Christine Jones and for giving us access to the SLINGbot to generate examples of botnet C2 traffic.

References

[1] Bot Roast II' Nets 8 Individuals. FBI Press Release; 2007 November 29.
[2] Over 1 Million Potential Victims of Botnet Cyber Crime. FBI Press Release; 2007 June 13.
[3] Symantec 2013 Internet Security Threat Report, vol. 18. Symantec Corp; 2013 April.
[4] Symantec Global Internet Security Threat Report, trends for July–December 07, vol. XIII. Symantec Corp; 2008 April.
[5] AsSadhan B, Moura JMF, Lapsley D, Jones C, Strayer WT, editors. Detecting botnets using command and control traffic. Proceedings of IEEE international symposium on Network Computing And Applications (NCA). Cambridge, MA, USA; 2009 Jul 9–11.
[6] AsSadhan B, Moura JMF, Lapsley D, editors. Periodic behavior in botnet command and control channels traffic. Proceedings of IEEE GLOBECOM. Honolulu, HI, USA; 2009 November 30–December 4.
[7] Gu G, Zhang J, Lee W, editors. BotSniffer: detecting botnet command and control channels in network traffic. Proceedings of the 15th annual Network and Distributed System Security Symposium (NDSS'08). San Diego, CA, USA; 2008 February 10–13.
[8] AsSadhan B, Kim H, Moura JMF, Wang X, editors. Network traffic behavior analysis by decomposition into control and data planes. Proceedings of international workshop on Security in Systems and Networks (SSN) in conjunction with the IEEE International Parallel and Distributed Processing Symposium (IPDPS). Miami, FL, USA; 2008 April 18.
[9] Oppenheim A, Schafer R, Buck J. Discrete-time signal processing. Upper Saddle River, New Jersey: Prentice-Hall; 1999.
[10] Priestley MB. Spectral analysis and time series. San Diego: Academic Press; 1981.
[11] AsSadhan B, Kim H, Moura JMF, editors. Long-range dependence analysis of control and data planes network traffic. Proceedings of the Saudi International Innovation Conference (SIIC). Leeds, UK; 2008 June 9–10.
[12] Ahdesmäki M, Lähdesmäki H, Yli-Harja O, editors. Robust fisher's test for periodicity detection in noisy biological time series. Proceedings of IEEE international workshop on Genomic Signal Processing and Statistics (GENSIPS). Tuusula, Finland; 2007 June 10–12.
[13] Thomson DJ, Lanzerotti L, Medford L, Maclennan C, Meloni A, Gregori G. Study of tidal periodicities using a transatlantic telecommunications cable. Geophys Res Lett 1986;13(6):525–8.
[14] Partridge C, Cousins D, Jackson AW, Krishnan R, Saxena T, Strayer WT, editors. Using signal processing to analyze wireless data traffic. Proceedings of ACM Workshop on Wireless Security (WiSe). Atlanta, GA, USA; 2002 September 28.
[15] Welch PD. The use of the fast Fourier transform for estimation of spectra: a method based on time averaging over short, modified periodograms. IEEE Trans Audio Electroacoustics 1967;15(2):70–4.
[16] So HC, Chan YT, Ma Q, Ching PC. Comparison of various periodograms for sinusoid detection and frequency estimation. IEEE Trans Aerospace Electron Syst 1999;35(3):945–52.
[17] Fukunaga K. Statistical pattern recognition. 2nd ed. San Diego: Academic Press; 1990.
[18] Trees HV. Detection, estimation, and modulation theory. Part 1. New York: John Wiley & Sons; 1968.
[19] Leon-Garcia A. Probability and random processes for electrical engineering. 2nd ed. Reading, Massachusetts: Addison Wesley; 1994.
[20] Brockwell P, Davis R. Time series: theory and methods. 2nd ed. New York: Springer-Verlag; 1991.
[21] Jackson AW, Lapsley D, Jones C, Zatko M, Golubitsky C, Strayer WT, editors. SLINGbot: a system for live investigation of next generation botnets. Proceedings of Cybersecurity Application and Technologies Conference for Homeland Security (CATCH). Washington, DC; 2009 March 3–4.
[22] LBNL/ICSI Enterprise Tracing Project. <http://www.icir.org/enterprise-tracing/> [accessed January 2013].
[23] Pang R, Allman M, Bennett M, Lee J, Paxson V, Tierney B, editors. A first look at modern enterprise traffic. Proceedings of ACM SIGCOMM/USENIX internet measurement conference; Berkeley, CA, USA; 2005 October 19–21.