Peer-to-Peer Netw. Appl.
DOI 10.1007/s12083-017-0586-0

Analysis of P2P, IRC and HTTP traffic for botnets detection

Basil AsSadhan[1] · **Abdulmuneem Bashaiwth**[2] · **Jalal Al-Muhtadi**[3] · **Saleh Alshebeili**[4]

Received: 16 November 2016 / Accepted: 6 July 2017

Abstract Botnets are widespread and have become a major threat to network security. A botnet is a group of infected computers that are controlled by a botmaster. Botnet's members use command and control (C&C) channels to communicate with their C&C server. In this paper, we study the detection of botnets by monitoring and analyzing botnets' C&C channels communication traffic. As bots are preprogramed to communicate every T seconds, we exploit this periodic behavior of C&C traffic to detect the botnet. The botnet detection approach we use is based on evaluating the periodogram of several count-feature sequences of the traffic and testing the significance of the peak of each periodogram. We apply this approach to real traffic that we captured from King Saud University's (KSU) network. The captured traffic contains more than 11 TB of traffic that spans 50 days during 2012 and 2013 from different locations inside KSU. We apply the detection approach to KSU's traffic to detect botnet C&C traffic that uses P2P, IRC, or HTTP as its communication protocols. The results show that the botnet detection approach can efficiently detect botnet members in recent traffic datasets. The period values of the detected bots ranged between 31 and 49 min.

Keywords Botnet C&C traffic detection · Peridoic behavior · Periodogram · IRC · P2P · Http

✉ Abdulmuneem Bashaiwth
basha429108167@gmail.com

Basil AsSadhan
bsadhan@ksu.edu.sa

Jalal Al-Muhtadi
jalal@ksu.edu.sa

Saleh Alshebeili
dsaleh@ksu.edu.sa

[1] Department of Electrical Engineering, Center of Excellence in Information Assurance (CoEIA), King Saud University, Riyadh, Saudi Arabia

[2] Department of Electrical Engineering, King Saud University, Riyadh, Saudi Arabia

[3] Department of Computer Science, Center of Excellence in Information Assurance (CoEIA), King Saud University, Riyadh, Saudi Arabia

[4] Department of Electrical Engineering, KACST-TIC in RF and Photonics for the e-Society (RFTONICS), King Saud University, Riyadh, Saudi Arabia

1 Introduction

A botnet is a group of compromised computers (*bots*) that are controlled remotely by a single entity called a *botmaster*. Botnets differ from other malware in the existence of command and control (C&C) channels. The botmaster uses C&C channels to communicate with botnet's members to receive commands, update software, and send keep-alive massages [1, 2].

Botnets are currently one of the major security threats that Internet users face. They are used to execute various malicious activities such as identity spoofing, password guessing, eavesdropping, DNS poisoning, Distributed Denial of Service (DDoS) attacks, E-mail spam, and phishing. Consequently, the detection of botnets has become an aim for network security administrators.

Most botmasters focus on making money through illegal means. The U.S. Federal Bureau of Investigation (FBI) in collaboration with Microsoft and the Financial Services Industry managed to disable more than 1000 botnet used for global cybercrime operation [3]. Financial Services Industry

Published online: 26 July 2017

has estimated the losses from that crimes to be more than half a billion U.S. dollars [4]. According to the 2014 security threats report from SOPHOS, credential-stealing botnet kit was used to steal over $250 million from financial institutions and their customers in middle of 2013 [5].

Botnets can be classified into two types, commonly called first and new generations [6, 7]. The first generation uses the *Internet Relay Chat* (IRC) and *Hypertext Transfer Protocol* (HTTP) services for its C&C communication channel. The use of HTTP in botnet C&C communication has increased in recent years as malware developers have moved beyond the IRC botnet of malicious bots. Using HTTP offers -from the attacker's perspective- legitimate manner since it is a widely-used protocol and firewalls seldom block it [8]. It also offers the advantage of stealth due to the high number of HTTP packets used in web browsing, which overshadow the low-volume C&C traffic, and make it difficult to detect.

IRC botnets while not growing rapidly, their numbers continue at a steady rate [9]. IRC botnets have integrated a couple of new features over the past years that make them stronger with higher level of threat than before [10]. In addition, the bots in IRC botnet respond faster to the botmaster because the push mode allows them to remain connected with their C&C servers and actively respond to botmaster commands [11]. Zhuge et al., [12], states that about 36% of discovered botnets use the standard IRC port 6667 to host C&C channels.

The first generation of botnets is of a centralized mechanism. The centralized C&C mechanism of such botnet has made it vulnerable to being detected and disabled because it has a single point of failure [13]. The new generation of botnets is *Peer-to-Peer* (P2P) based botnets [14]. The P2P botnet does not suffer from a single point of failure, as it does not have a centralized C&C server [13, 15]. Instead, botnet's members contact each other through a mesh topology [7, 16]. P2P bots use port numbers ranging from 10,000 to 30,000 in their communication [17].

Regardless of the different structures and communication protocols used in several botnet variants, bots within a single botnet frequently contact each other through C&C communication channels every T seconds to receive commands, update data, and send keep-alive messages. Due to this behavior, these bots demonstrate similar traffic activities which result in temporal-spatial correlation [18]. Therefore, and as result of the pre-programmed manner in bots, periodic behavior arises in botnet C&C channels traffic. In our work, we exploit this behavior by analyzing network traffic to detect botnets.

In order to analyze network traffic, we need first to obtain packet traces. There are several approaches to obtain packet traces, and these traces differ in their characteristics depending on the approach used in obtaining them. Packet traces can be obtained by; 1) Generating traffic in an isolated experimental network environment (e.g., [19]); 2) Simulating network traffic (e.g., [20]); 3) Capturing traffic from real world networks (e.g., [21]). Both generating traffic in an isolated experimental network environment and simulating it provide packet traces that contain header and payload information. Network traffic generated in an isolated network is a better representation of real world network traffic than simulated traffic. However, it is not easy to either simulate or generate network traffic with all of the network applications that real world networks have. Therefore, generated network traffic is only a good representation of one or a group of applications traffic.

Among the options discussed above, we resort to capturing packet traces from King Saud University's (KSU) network. We preprocess these packet traces to extract packet and address count sequences over a suitable aggregation interval. This enables us to examine the periodic behavior of the extracted sequences to detect botnet C&C channels traffic. We use KSU's dataset, because it contains real (not simulated) and relatively recent Internet traffic. In addition, this dataset reflects newer traffic patterns, with focus on social media, online streaming and other newer applications. So our contributions in the paper include:

- Capturing our own dataset, and preprocessing it to extract the needed count-feature sequences to enable the use of our proposed detection method.
- Validating the proposed method by detecting the periodic behavior of botnet C&C traffic in this dataset. Along with the discussion that the address count sequences are more robust than the packet count sequences and they can resist the presence of background traffic.
- Demonstrating how can we use the proposed method to detect HTTP botnet C&C traffic by filtering out irrelevant HTTP traffic and focusing only weekend traffic.

The rest of this paper is organized as follows; in Section 2, we review related work in botnet detection. Section 3 describes the methodology we adopt to test the presence of periodic behavior in network traffic. Section 4 lists our experimental work on botnet detection, which includes how we capture and preprocess packet traces. Section 5 presents and evaluates the results of the approach on various network traces. We present our conclusions in Section 6.

2 Background: detection of botnet C&C traffic

Research on botnet detection mainly focuses on three issues; detection of botnet itself, bots tracking, and defending against botnet attacks. The goal of detection of botnets is to know whether a host is remotely controlled or not; while the purpose of tracking bots is to detect more bots and even find the botmaster. Defending against botnet attacks aims to block the communication between the bots and their botmaster [22].

Yu et al., [23], present a method to detect botnet based on similarity measurement. This is achieved by computing the average Euclidean distance between streams of host features. Once few feature streams exhibit high similarity in their activities, the corresponding hosts are regarded as suspected bots. The authors extract several features to construct the feature stream, example of these features include; total packet exchanged in flow, average bits per second for flow, and flow duration. In their work, they use the Discrete Fourier Transform (DFT) to avoid huge calculation among feature streams.

Arshad et al., [24], follow a similar approach to the one used by Yu et al. where they measure the similarity between NetFlows [25], to detect botnet. They monitor the behavior of NetFlows and attacks simultaneously. As bots connect to the C&C channel and execute the received commands, bots belonging to the same botnet receive the same commands. Consequently, the bots have similar Netflow's characteristics and perform the same attacks. To identify which hosts are responsible for botnet, they cluster bots with similar NetFlows and attacks in different time windows, and then perform correlation techniques. The method is applied to real-world packet traces including normal traffic and several real-world botnet traces that include IRC-SdBot, IRC-SpyBot, HTTP-Bot-I, and HTTP-Bot-II.

To detect IRC botnet C&C traffic and distinguish it from IRC chat traffic, Ma et al., [26], propose to analyze the characteristic of packet size sequences of the TCP conversation held between IRC bots and their C&C servers. They report that the TCP conversations within IRC botnet show a nature of approximate (quasi) periodicity, whereas the ones in IRC chat do not show periodicity. Ukkonen algorithm was used to measure the quasi-periodicity degree in IRC conversations. On the other hand, AsSadhan and Moura [27], study the periodic behavior in C&C traffic, they evaluate the periodogram of the traffic, then apply Walker's large sample test to detect whether the traffic has a significant periodic component or not.

Goebel et al., [28], report that the infected machines belong to an IRC botnet often have nicknames different from that of normal machines. They use a scoring system to detect bots that use uncommon communication channels. The scoring function checks for the occurrence of several criteria monitored using Rishi tool such as suspicious substrings, special characters, or long numbers. For each successful test, a certain number of points are added to the overall score that the particular nickname has already received. The higher the score a nickname receives, the more likely it is a bot infected machine trying to contact its C&C server.

Most reviewed techniques share the following characteristics; 1) they are independent of the structure and communication protocol used in the botnet, 2) they look for periodic behavior or similar activities in botnet traffic, 3) they do not need any other a priori knowledge of the botnet behavior. Furthermore, Table 1 provides comparison between the reviewed techniques. It is clear that most of the reviewed techniques work in the time domain and concentrate on packet counts as the extracted feature.

Our proposed method differs than most of the reviewed techniques in two parts; first the analysis of the traffic is done in frequency domain. This involves less amount of computations, as these computations depend on the use of Fast Fourier Transform (FFT) [29]. Second, the method extracts both features, packet and address counts, where we show in our results that the address count sequences are more robust than the packet count sequences and they can resist the presence of background traffic. We next describe our methodology to detect periodic behavior in botnet C&C traffic.

3 Botnet detection methodology

We detect botnet C&C traffic through the detection of periodic behavior in network traffic. This is done by analyzing the Power Spectral Density (PSD) of the count-feature sequences extracted from the traffic. The flow chart in Fig. 1 presents the detection methodology we use. The first step is to extract the count feature sequences which we then estimate their PSD using Periodograms [30]. After that, we determine if the count feature sequence exhibits periodic behavior or not. If it does, we need to validate that this behavior comes from bot traffic. A quick step to validate whether the traffic comes from a bot is to check what port number it is sent on. If the port number is among the suspected port numbers used by the botmaster such as 6667 and 11,375, then this is a high indication that the traffic is generated by a bot.

The detection of botnet traffic in our work is based on exploiting periodic behavior of C&C traffic. Exploiting the periodic behavior in traffic to detect bots was performed in previous studies [18, 27]. The autocorrelation function of the host's traffic in [18] was computed in the time domain to test the presence of periodic component. In our work, the presence of periodic components in the traffic is examined in frequency domain through analyzing PSD of this traffic. One of the most used tools to estimate the PSD of signals are periodograms. The periodogram of a time sequence (signal) provides its power at different frequencies [30].

The frequency components of a periodic signal exhibits high level of power at its fundamental frequency. Therefore, the periodogram of a periodic signal will have a high peak located at the fundamental frequency of the signal when compared to the mean of the periodogram. We extract count-feature sequences over a selected aggregation interval to produce a discrete time sequence $x[n]$. Then we calculate the periodogram of these sequences. The periodogram, $P_{xx}[k]$, of a discrete time sequence $x[n]$ is defined as the square

Table 1 Summary of reviewed detection methods in this paper

Method	Detection Algorithm	Dataset	Features	Domain	Detection Approach	Independence of C&C Protocols	Need of a priori knowledge of botnet behavior
Gu et al., [18]	Spatial-Temporal Correlation	Traces from University's Network	Packet Counts	Time	Similarity	Yes	No
Yu et al., [23]	Discrete Fourier Transformation (DFT)	IRC-sdbot IRC-agobot IRC-rbot P2P-nugache	Packet and Byte counts	Frequency	Similarity	Yes	No
Arshad et al., [24]	Clustering and Correlation	IRC-SdBot HTTP-Bot	Packet and Byte counts	Time	Similarity	Yes	No
Ma et al., [26]	Ukkonen Algorthim	Honeynet Traces	Packet size	Time	Periodicity	Yes	No
AsSadhan et al., [27]	Power Spectral Density (PSD)	SLINGbot & LBNL/ICSI Traces	Packet and IP address Counts	Frequency	Periodicity	Yes	No
Goebel et al., [28]	Rishi tool and Scoring function	RWTH Aachen University Network	IP address and Port number	Time	Abnormality	Yes	Yes

magnitude of the Discrete Fourier Transform (DFT) of the signal. It is evaluated by

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

where

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp\left(\frac{-j2\pi kn}{N}\right)$$
$$P_{xx}[k] = \frac{1}{N}|X[k]|^2, \tag{2}$$

is the N- point DFT.

After evaluating the periodogram of the signal and locating its peak, we test the significance of the peak compared to the mean of the periodogram's ordinates. AsSadhan and Moura [27], set the following statistic,

$$g_x^* = \frac{\max_{0 \le k \le m-1}(P_{xx}[k])}{\frac{1}{2m}\sum_{k=0}^{m-1}(P_{xx}[k])}. \tag{3}$$

The sample ratio test statistic g_x^* in (3) represents the maximum value of periodogram divided by the mean of periodogram's ordinates. This ratio will have a high value when the signal is periodic and a low value when the signal is aperiodic signal. AsSadhan and Moura show that the distribution of g_x^* when the signal is aperiodic can be represented as a maximum function of m exponential distributions with mean 2 [27], where m is is the number of ordinates at the positive frequencies of the periodogram $P_{xx}[k]$. Therefore, it follows that for $z_\alpha \ge 0$,

$$\Pr[g_x^* > z_\alpha] \sim 1 - \left(1 - \exp\left(-\frac{z_\alpha}{2}\right)\right)^m. \tag{4}$$

We select the false alarm probability α in (4) based on how small we want the probability of false alarm to be. A false alarm probability here refers to declaring that a signal is periodic, because of g_x^* being larger than z_α, where in fact it is aperiodic. This results in the following threshold to test the peak's significance,

$$z_\alpha = -2\ln\left(1 - (1-\alpha)^{1/m}\right). \tag{5}$$

If g_x^* is larger than z_α, we conclude that the sequence has a periodic component with a false alarm probability of α, and

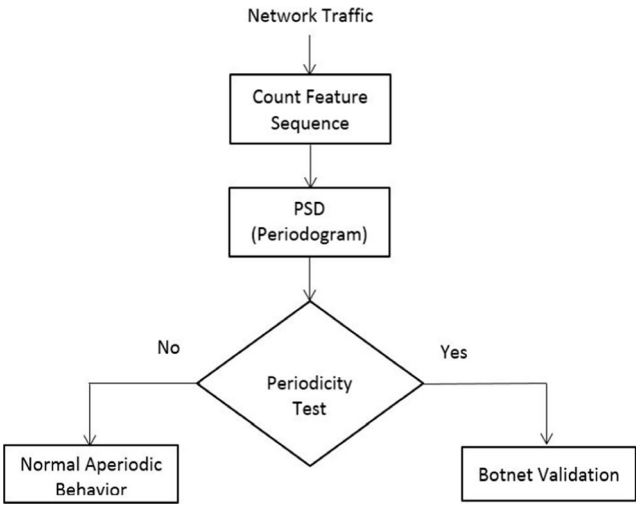

Fig. 1 The flowchart of the detection methodology

that the periodic behavior is due to bot C&C traffic. If g_x^* is less than z_α we conclude that the sequence does not have a periodic component.

In certain circumstance, some of the harmonic components in a periodogram can have a higher peak than the one located at the fundamental frequency. To address this issue, AsSadhan [31], proposed an extension of the test to detect the fundamental frequency's peak. It is based on an iterative process that is applied after evaluating the periodogram to identify the lowest frequency in the periodogram where a significant peak is located.

4 Experimental setup

4.1 Capturing network traffic

To capture network traffic, we use Endace DAG 7.5G2 card [32]. The Endace card is used to capture traffic at the full line rate from the network into the memory of the host computer with zero packet loss. The card[1] operates on a 4 lane PCIe bus and can be installed in any free 4 lane PCIe slot [33].

The DAG 7.5G2 card provides two Gigabit Ethernet interfaces. The interfaces can come as either optical or copper depending on the underlying network. It is capable of capturing full line rate (1 Gbps) of Ethernet traffic. Capturing packet at full line rate allows recording all header information and/or payload with a high precision timestamp. Presence of two interfaces allows simultaneously capturing traffic from different sources (e.g., switches or routers).

The DAG card produces trace files in its own native Endace Record Format (ERF). ERF files contain a hardware generated timestamp of each packet's arrival. The format of this timestamp is a single little-endian 64-bit fixed point number, representing the number of seconds since Epoch time (i.e., midnight on the January 1st, 1970). The most significant 32-bits represent the integer number of seconds, while the least significant 32-bits represent the binary fraction of the second. This allows an ultimate resolution of 2^{-32} s, or approximately 233 ps.

4.2 Capturing scheme

Fig. 2 shows how to capture a router's network traffic by mirroring the traffic of one or more of its ports into one of the DAG7.5G2 card ports through the *Switched Port Analyzer* (SPAN) of the router. The captured traffic is stored in the hard drive of the PC hosting the card.

By this scheme, we managed to capture more than 11 TB worth of traffic using DAG 7.5G2 in native ERF format to cover all hours of the day for 50 days starting from Saturday December 22, 2012 until Saturday February 9, 2013. Approximately, more than 10,000 hosts were active inside KSU network during the capturing process. These hosts were from more than 50 different locations including labs, offices, student housing, and university hospital, which makes the captured traffic diverse.

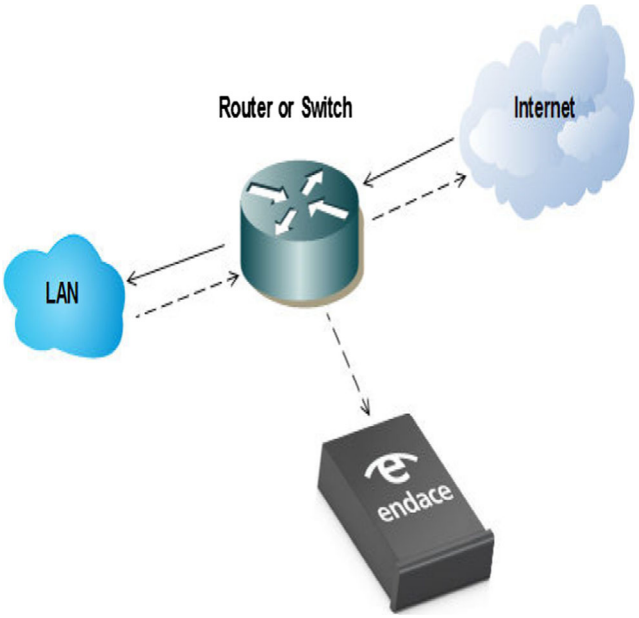

Fig. 2 The capturing scheme, which shows how to connect the DAG card to the router's SPAN port

4.3 Anonymization of packet traces

To protect the identity of user's private information, IP addresses have to be anonymized and only packets' header information can be publicly released. In our work, we anonymize the IP addresses of KSU packet traces, where each distinct IP address appearing in the original trace is mapped to a distinct random address, thus the mapping process is performed one-to-one. The address anonymization technique is prefix-preserving, where two IP addresses sharing an n-bits prefix in the original IP address space will also share an n-bit prefix in the anonymized IP address space [34]. Preserving the prefix of IP address guarantees the hosts in the same subnet in the original IP address space to be in the same subnet in the anonymized IP address space.

We use the Cryptography based *Prefix preserving Anonymization* (Crypto-PAn) tool to perform IP address anonymization [35]. The Crypto-PAn tool is used to perform prefix-preserving anonymization, which uses the IP::Anonymous built-in Perl's module [36]. One of the important characteristics of Crypto-PAn is that it uses cryptography techniques that allow the owner of the traces to use a secret key to keep the anonymization process secret. The following example explains the idea of IP address anonymization we

[1] Endace card works on Linux, FreeBSD, Windows Server 2003/2008, and Windows 7–64 bit operating systems

use. The length of preserved prefix in this example is 16 bits as shown in the underlined part of bit sequences.

Original IP addresses:

IP1: 10.16.3.5 (00001010.00010000.00000011.00000101)
IP2: 10.16.220.3 (00001010.00010000.11011100.00000011)

Anonymized IP addresses:

IP1*: 117.12.14.250 (01110101.00001100.0001110.11111010)
IP2*: 117.12.92.115 (01110101.00001100.01011100.01110011)

4.4 Traffic Preprocessing

We use Perl [37], to convert captured packets into Comma Separated Value (CSV) files and anonymize IP address of these packets simultaneously. We start by recognizing the timestamp and data length fields between packets. Then we read the Ethernet frame header for each packet, where we can recognize IP packets. After that, by parsing the IP header, we were able to get TCP packets. We focus on TCP packets since they constitute 85–90% of Internet's traffic. At KSU's dataset, we found that TCP packets represent more than 90% of IP packets. For each TCP packet, we extract the following information and store them in CSV files:

- Time stamp
- Source IP address
- Source port
- Destination IP address
- Destination port
- Total length of the frame
- TCP flag (SYN, FIN, RST, or no flag is set)
- Data sequence number of the packet
- Data sequence number of the data expected in return
- Acknowledgment sequence number of the expected next data
- Receiver window size i.e., the number of bytes that receiver can receive
- Total length of the frame

We use the first six of the above features to produce discrete-time sequences. This is performed by aggregating TCP packets over an appropriate aggregation interval. Then count-features sequences (packet, byte, address, and port counts) are extracted from TCP packet header information. As we notice that the byte count sequences follow packet count sequences, and the port count sequences are similar to the address count sequences, we will concentrate only on packet and address counts sequences. Having discrete-time sequences enable us to use statistical signal processing techniques such as periodograms for the purpose of botnet detection.

5 Results and discussion

In this section, we test our botnet detection approach to detect the periodic behavior of botnet C&C traffic. In our analysis, we adopt a time window of 24 h for analyzing traffic. We select this duration to cover all hours of the day, as the amount of traffic during busy hours of the day is high and can overshadow the periodicity of botnet traffic. On the other hand, analyzing night hours increases chances of detection as they have less amount of background traffic. We note that, if the botnet uses a period that is larger than the duration of analyzed traffic, then we will not be able to detect it.

During our analysis of KSU's captured traffic, we face the problem of the huge amount of traffic. There were more than 10,000 active hosts inside KSU network that were communicating using several different port numbers. Since checking each host's traffic for periodic behavior is not feasible, we resort to a more scalable approach. Two approaches can be used to decrease the amount of traffic to be analyzed; the first approach is to analyze the traffic on all communication port numbers for a given host or a given group of hosts (e.g., subnet) to detect periodic behavior. The second approach is to analyze the traffic of all hosts for each (service port number), separately. In both approaches, if periodic behavior is detected in the subnet/port number, we can then search for the responsible host for that periodic behavior traffic within the subnet/port number.

We found that the first approach was not successful in detecting the periodic behavior of C&C traffic even when only analyzing the traffic of one host during weekend night. This is due to the very low volume of botnet C&C traffic that is overshadowed by the rest of host's traffic. Therefore, we adopt the second approach in our analysis of the traffic. Based on prior knowledge that the botmaster usually performs C&C communication on certain port numbers, we apply the second approach to filter the traffic of three port numbers; P2P traffic on port numbers 11,375, IRC traffic on port number 6667 and HTTP traffic on port number 80. We present and discuss results on each of these port numbers next.

5.1 P2P botnet traffic on port number 11375

In this subsection, we attempt to detect C&C traffic in KSU's network by applying the detection approach. We analyze the traffic that originates from or is directed to KSU's hosts on port number 11375 during a 24-h window starting at 6 AM on a given weekend day (i.e., Thursday or Friday[2]) until 6 AM on the next day. We select weekend days, since their traffic have

[2] During the capturing of the traffic, Thursday and Friday were the weekend days in the Kingdom of Saudi Arabia.

low volume during day hours. We avoid week days, since their traffic have high volume during busy hours. We note that high volume traffic can overshadow the low-volume C&C traffic, and negatively affects the chances of detecting its periodic behavior.

The top plots of Fig. 3 show the packet and address count sequences for the packet traces captured from KSU network on Thursday December 27, 2012 on port number 11375. An aggregation interval of 60 s is used to extract the two count sequences from this packet traces. The bottom plots in the same fig. Show the periodogram for each sequence after subtracting its mean and normalizing it by its standard deviation. As can be seen from the bottom plots, both periodograms have a maximum peak located at 1 mHz. Although, this peak is the maximum but is located at one of the harmonics, and not at the fundamental frequency. However, the peak located to the left of the maximum peak at 0.5 mHz is also detected. This is due to the execution of the extension technique of testing the significance of periodogram's peak pointed to in Section 3. The value of the ratio test statistic g_x^* are 29 and 38 for the periodograms of packet and address count sequences, respectively.

The detected peak at 0.5 mHz is significant as the values of g_x^* are larger than the value of the threshold $z_{1\%}$, which is equal to 23.1[3] at 1% false alarm probability. Therefore, the packet and address count sequences in Fig. 3 exhibit periodic behavior at 0.5 mHz, which corresponds to a period of 33.3 min. There is a tradeoff in the selection of the false alarm probability between raising the detection rate of periodic behavior and reducing the false alarm rate (i.e., declaring that the traffic has periodic behavior where in fact it does not). We select a value of 1% for the false alarm probability to reduce the false alarm rate, as it is used by others.

From the top plots of Fig. 3, we note that the C&C traffic is active for a very short time during the period duration of 33.3 min, which results in a very low duty. AsSadhan and Moura [27], report that the ratio between the periodogram's main peak and the harmonic components is higher in periodic signals with higher duty cycle. This means that network traffic that exhibits periodic behavior with low duty cycle produces smaller g_x^* than that produced from network traffic with high duty cycle. The reason of the low duty cycle in C&C traffic in Fig. 3 is that the C&C traffic exchange involves very few packets that only represent the traffic originating from KSU's hosts, whereas the response to these packets does not appear. This is believed to be due to a firewall configuration by KSU's network administrator, where incoming traffic on port numbers such as 11,375 is blocked.

[3] The number of ordinates at the positive frequencies of periodogram, m, used in evaluating $z_{1\%}$ is 1024.

The top plots of Fig. 4 show the separated sequences of outgoing and incoming packet count sequences in Fig. 3.

The outgoing packets are the packets that originated by KSU's network hosts, while the incoming packets are that their destination is KSU's network hosts. As it can be seen from Fig. 4, almost all the packet count sequences on port 11,375 are due to the outgoing traffic, whereas the incoming traffic has very few packets and its periodogram does not have any peaks. Therefore, we can conclude that the periodic behavior of the packet and address count sequences in Fig. 3 is due to an internal bot issuing C&C traffic in the outgoing traffic.

Next, we identify which KSU host was responsible for the periodic behavior. This is done by searching through all KSU's hosts and observing the number of transferred packets on port 11,375. The host(s) that generates a significant amount of packets when compared to the number of all transferred packets is treated as suspicious, and their traffic is further analyzed. After performing this search, we found that an internal host is responsible for the C&C traffic on port number 11375. The anonymized IP address of this host is 245.230.103.2. Since this suspicious host appears several times in our analysis, we name it Sohail so that it will be easier to refer to it.

The top plots of Fig. 5 show the packet and address count sequences of Sohail on port number 11375 at the same day (i.e., Thursday December 27, 2012). In both periodograms a maximum peak is located at the same frequency (0.05 mHz) of periodograms in Fig. 3. The values of the ratio test statistic g_x^* are 44 for both We use the same aggregation interval of 60 s and extract the packet and address count sequences within this aggregation interval. Periodograms of packet and address count sequences. The bottom plots show the periodogram for each sequence after subtracting its mean and normalizing it by its standard The maximum peak at this frequency is significant as the values of g_x^* are larger than the value of the threshold $z_{1\%}$ (23.1). Therefore, the packet and address count sequences in Fig. 5 exhibit periodic behavior at 0.5 mHz, which corresponds to a period of 33.3 min.

Based on this result, we can conclude that the KSU's network traffic on port number 11375 has botnet C&C traffic that exhibits periodic behavior at 33.3 min, and conclude that Sohail is responsible for that C&C traffic. We also notice that the values of the ratio test when testing Sohail traffic alone (44) are higher than the values of the test ratio when testing the traffic of all internal hosts in the network (29 and 38 for packet and address count sequences, respectively). This is due to the presence of legitimate background traffic with the bot's C&C traffic in case of testing the traffic of all internal hosts on port number 11375, which reduces the effect of periodic behavior.

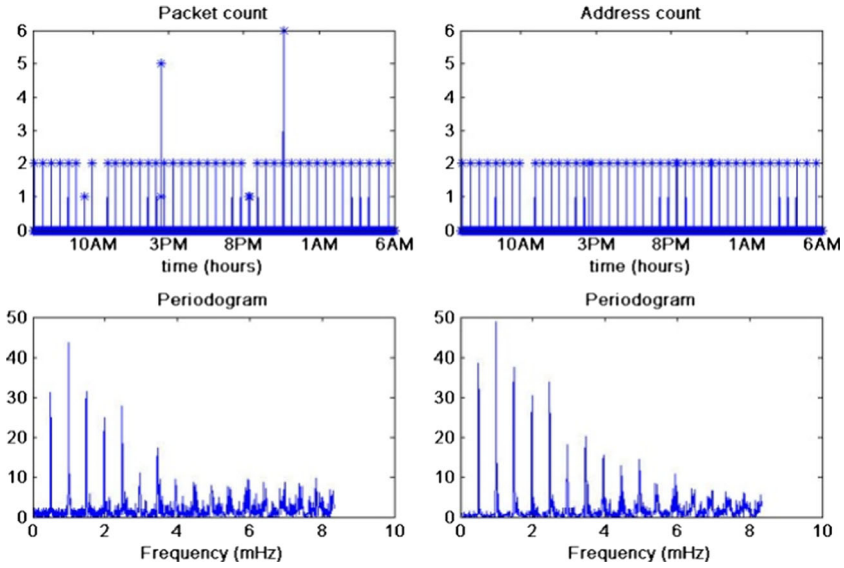

Fig. 3 Packet and address count sequences and their one sided periodogram for P2P traffic captured from KSU network on Thurs. Dec. 27, 2012 on port 11,375 using an aggregation interval of 60 s

5.2 IRC botnet traffic on port number 6667

In this subsection, we use the botnet detection approach to detect the periodic behavior in IRC KSU's network traffic on port number 6667. We analyze the traffic that originates from or is directed to KSU's hosts on port number 6667 during a 24-h window starting at 6 AM on a given day until 6 AM on the next day. The top plots of Fig. 6 show the packet and address count sequences of IRC traffic for the packet traces captured from KSU network on Thursday December 27, 2012. An aggregation interval of 60 s is used to extract the two count sequences from IRC traffic. The bottom plots in the same fig. Show the periodogram for each sequence after subtracting its mean and normalizing it by its standard deviation.

In both periodograms, a maximum peak is located at 0.34 mHz. The values of the ratio test statistic g_x^* are 24 and 43 for the periodograms of packet and address count sequences, respectively. The maximum peak at 0.34 mHz is significant as the values of g_x^* are larger than the value of the threshold $z_{1\%}$ (23.1). Therefore, the packet and address count sequences in Fig. 6 exhibit periodic behavior at 0.34 mHz, which corresponds to a period of 49 min.

From the top plots of Fig. 6 we note that, as in the traffic on port 11,375, the C&C traffic is active for a very short time during the period duration of 49 min. This is because the C&C traffic exchange involves very few packets that only represent the traffic originating from KSU's hosts, whereas the response to these packets never appears. As explained earlier the reason behind this is believed to be due to a firewall configuration that blocks incoming traffic on port numbers such as 6667. Next, we identify which KSU host was responsible for the periodic behavior using the same searching mechanism used

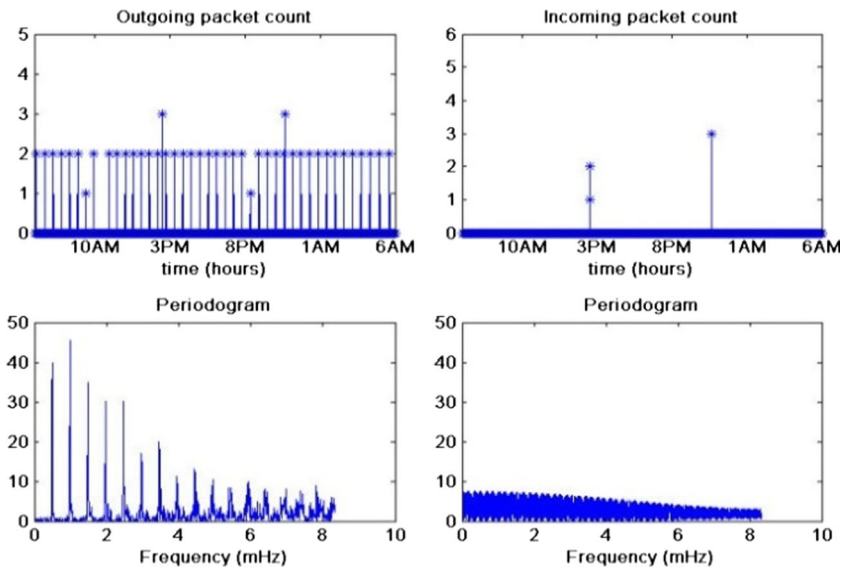

Fig. 4 Outgoing and incoming count sequences of the packet count sequences shown in Fig. 3 and their one sided periodogram

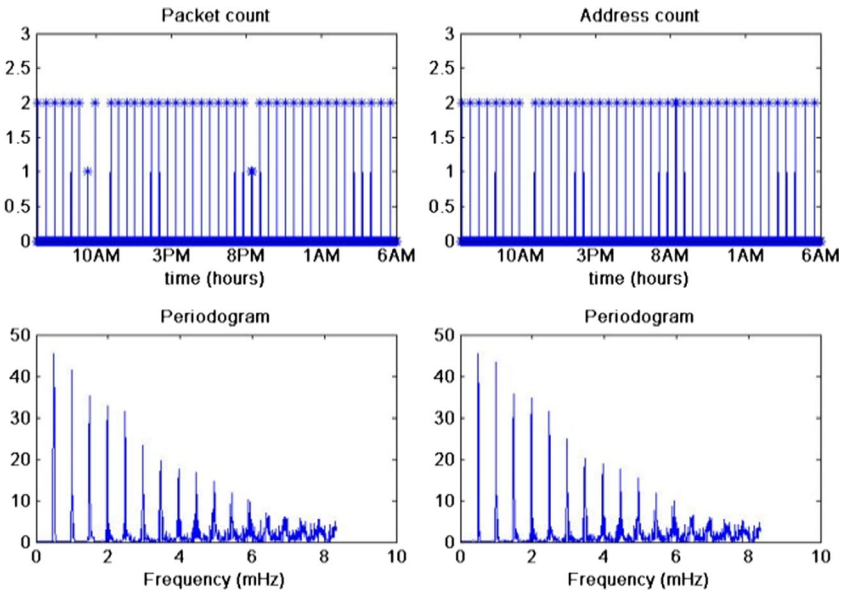

Fig. 5 Packet and address count sequences and their one sided periodogram for P2P traffic captured from Sohail on Thurs. Dec. 27, 2012 on port 11,375 using an aggregation interval of 60 s

in Subsection A. After performing this search on the traffic exchanged on port 6667, we found that Sohail is again responsible for the C&C traffic.

The top plots of Fig. 7 show the packet and address count sequences of Sohail on port number 6667 at the same day (Thursday December 27, 2012). We use the same aggregation interval of 60 s and extract the packet and address count sequences within this aggregation interval. The bottom plots show the periodogram for each sequence after subtracting its mean and normalizing it by its standard deviation. We can see a maximum peak is located at the same frequency (0.34 mHz) in Fig. 6. The values of the ratio test statistic g_x^* are 45 for both periodograms of packet and address count sequences. The maximum peak at this frequency is significant as the values of g_x^* are larger than the value of the threshold $z_{1\%}$ (23.1). Therefore, the packet and address count sequences in Fig. 7 exhibit periodic behavior at 0.34 mHz, which corresponds to a period of 49 min.

Based on this result, we can conclude that KSU's network traffic on port number 6667 has botnet C&C traffic that exhibits periodic behavior at 49 min, and conclude that Sohail is responsible for that C&C traffic. We also notice that the values of the ratio test when testing Sohail traffic alone (45) are higher than the values of the test ratio when testing traffic of all internal hosts in the network (24 and 43 for packet and address count sequences, respectively). This is due to the

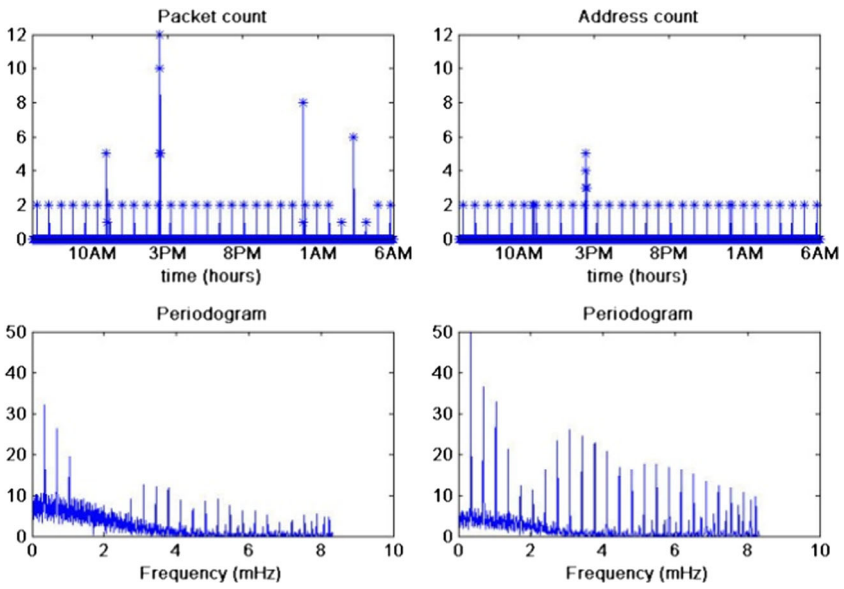

Fig. 6 Packet and address count sequences and their one sided periodogram for IRC traffic captured from KSU network on Thurs. Dec. 27, 2012 on port 6667 using an aggregation interval of 60 s

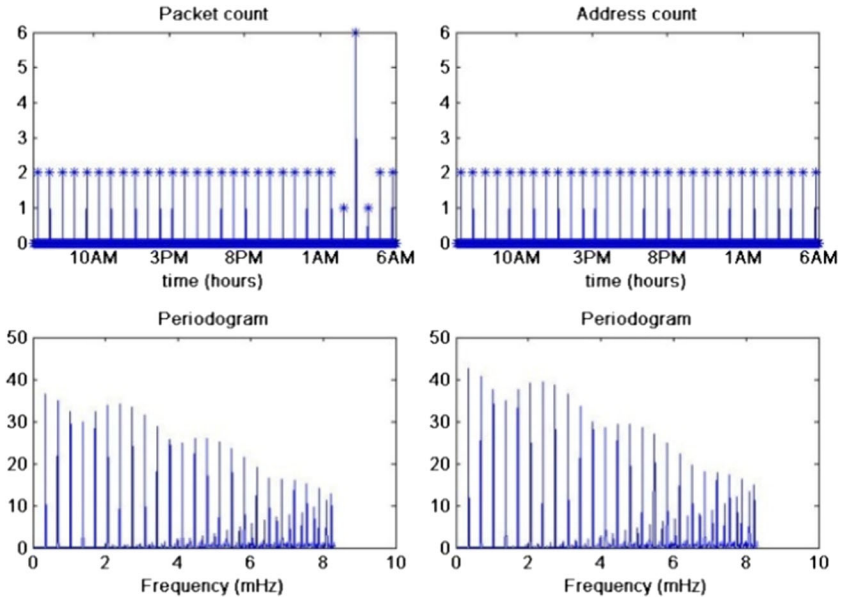

Fig. 7 Packet and address count sequences for IRC traffic and their one sided periodogram captured from Sohail on Thurs. Dec. 27, 2012 using an aggregation interval of 60 s

presence of legitimate background traffic with the bot's C&C traffic in case of testing the traffic of all internal hosts on port number 6667, which reduces the effect of periodic behavior.

5.3 HTTP botnet traffic on port number 80

In this subsection, we attempt to detect C&C botnet traffic in HTTP KSU's network traffic on port number 80. We tested HTTP traffic in a time window of 24 h or less, during time periods that are expected to have low amount of traffic (e.g., after midnight on weekend) to detect periodic behavior. The result of the test shows that the traffic does not exhibit periodic behavior. The examined traffic might have a periodic behavior, but the high amount of background traffic suppresses its effect. To address this issue, we decrease the amount of traffic we analyze by considering only HTTP traffic exchanged between one external host and all internal hosts of KSU network. In previous subsections, we detected that Sohail's network traffic exhibits periodic behavior on both port numbers 11,375 and 6667 that it is due to botnet C&C traffic. Consequently, Sohail is suspicious to be a member of HTTP botnet that communicates with external hosts. Therefore, we filter KSU's traffic on port 80 between Sohail and external hosts in order to detect presence of botnet C&C traffic. After filtering, we found that there are few external hosts that exchange a noticeable amount of traffic with Sohail. We applied the botnet detection approach to the filtered traffic; however, we did not observe any periodic behavior. We performed the test again, but this time on HTTP traffic between Sohail and each external host separately.

The result shows periodic behavior in HTTP traffic between Sohail and the external host with IP address (158.200.14.76). This host has high probability to be a HTTP C&C server as when we test the traffic behavior between this host and all KSU hosts, we find that the traffic exhibits periodic behavior whether we analyze its traffic with all KSU hosts together or separately.

The top plots of Fig. 8 show the packet and address count sequences of KSU's network traffic exchanging between all KSU hosts and external host (158.200.14.76) on port number 80 captured on Friday December 28, 2012. An aggregation interval of 60 s is used to extract the two count sequences from HTTP traffic. The bottom plots show the periodogram for each sequence after subtracting its mean and normalizing it by its standard deviation. In both periodograms, a maximum peak is located at 0.54 mHz. The values of the ratio test statistic g_x^* are 25 and 108 for the periodograms of packet and address count sequences, respectively. We 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 because the number of distinct addresses in the traffic flow has fewer fluctuations when compared to the number of packets. The maximum peak at 0.54 mHz is significant as the values of g_x^* are larger than the value of the threshold $z_{1\%}$ (23.1). Therefore, the packet and address count sequences in Fig. 8 exhibit periodic behavior at 0.54 mHz, which corresponds to a period of 31 min. Ten KSU hosts were connected to this external host, and the traffic of each one of them exhibits periodic behavior at the same period of 31 min.

Now, we show the traffic between 158.200.14.76 and only one of these KSU's hosts since the traffic of other hosts is very similar. The top plots of Fig. 9 show the packet and address count sequences of KSU's network traffic exchanging between KSU's host (245.223.234.223) and external host (158.200.14.76) on port number 80 captured on Friday December 28, 2012. An aggregation interval of 60 s is used

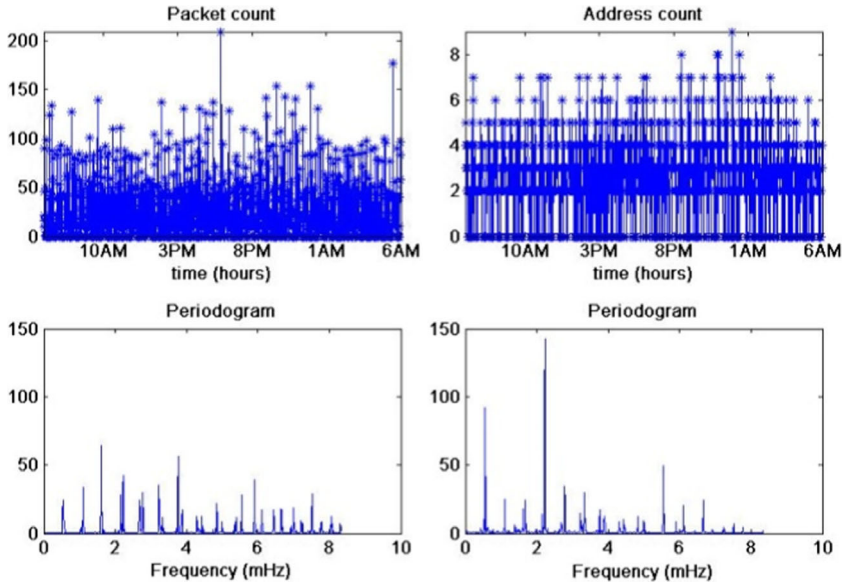

Fig. 8 Packet and address count sequences and their one sided periodogram for HTTP traffic exchanged between all KSU hosts and external host 158.200.14.76 captured on Fri. Dec. 28, 2012 using an aggregation interval of 60 s

to extract the two count sequences from HTTP traffic. The bottom plots show the periodogram for each sequence after subtracting its mean and normalizing it by its standard deviation. In both periodograms, a maximum peak is located at the same frequency (0.54 mHz) of periodograms in Fig. 8.

The values of the ratio test statistic g_x^* are 60 and 89 for the periodograms of packet and address count sequences, respectively. The maximum peak at 0.54 mHz is significant as the values of g_x^* are larger than the value of the threshold $z_{1\%}$ (23.1). Therefore, the packet and address count sequences in Fig. 9 exhibit periodic behavior at the same period (31 min) of the count sequences in Fig. 8. Based on this result, we can conclude that the KSU's network traffic on port number 80 has botnet C&C traffic that exhibits periodic behavior at 31 min.

5.4 Summary of results and limitations

We use the botnet detection approach to detect botnet C&C traffic that uses three different types of network protocols (P2P, IRC, and HTTP). We evaluated the botnet detection approach by applying it to real-world network traffic to detect possible actual botnet C&C traffic. We used traffic captured from KSU network in the evaluation, and we found that this traffic contains botnet C&C traffic which exhibits periodic behavior at periods of 33.3 min in case of 11,375 traffic and

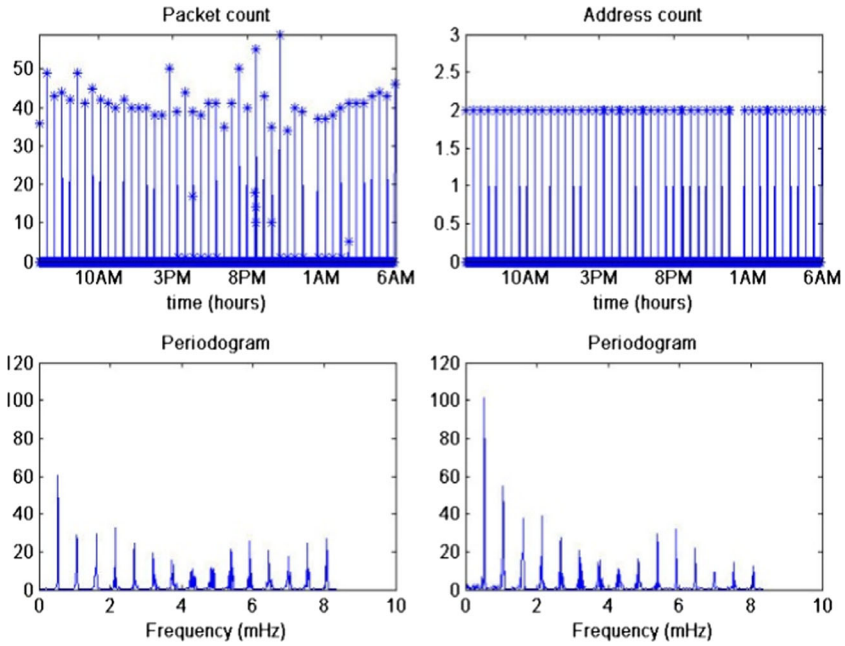

Fig. 9 Packet and address count sequences and their one sided periodogram for HTTP traffic exchanged between KSU host 245.73.244.243 and external host 158.200.14.76 captured on Fri. Dec. 28, 2012 using an aggregation interval of 60 s

49 min in case of 6667 traffic. In case of HTTP traffic on port 80, we are forced to decrease amount of HTTP traffic analyzed on this port.

It is difficult to detect periodic behavior in HTTP traffic due to the huge amount of traffic on this port that suppresses the effect of the periodic behavior. We used only the HTTP traffic exchanged between suspicious external host and all internal hosts at KSU network. The results show that this traffic has botnet C&C traffic which exhibits a periodic behavior at a period of 31 min.

In the next lines, we address some of the limitations of observing periodic behavior as a basis for the detection of botnet C&C traffic. In some botnets variants, the botmaster may attempt to avoid detection by uniformly randomizing the period in a given small range. This can be modeled as a random phase. The detection of periodic behavior in such case will depend on how large the random phase is and on the period's length and the duty cycle [27].

Using a larger range will damage the periodic behavior, and evades the detection. However, using such evasion scheme will limit the efficiency of the exchange of C&C channel traffic. As a result it may disturb the effectiveness of the attacks carried out by the botnet because bots are not receiving C&C updates at predetermined times.

6 Conclusions

In this paper, we analyze network traffic to detect botnet C&C traffic. To do that, we start by capturing traffic from KSU's network. We capture and preprocess more than 11 TB worth of traffic that cover all hours of the day for 50 days starting from December 2012 until February 2013.

We detect botnet C&C traffic in KSU's P2P, IRC and HTTP traffic over a time window that spans 24 h. Our detection approach is based on detecting the periodic behavior of C&C traffic. It was performed by analyzing the traffic of all hosts over port numbers 11,375 and 6667. Upon detection of periodic behavior, we search for the responsible host for that periodic behavior. Our results point to the existence of periodic behavior that represents botnet C&C traffic in P2P and IRC traffic. Moreover, we are able to determine the responsible bot for C&C traffic as shown in Figs. 3, 4, 5, 6 and 7. The periods of the bot were; 33.3 min, and 49 min for P2P and IRC traffic, respectively.

We attempted to detect botnet C&C traffic in HTTP traffic on port number 80 for all hosts. Unfortunately, since we have a huge amount of HTTP traffic on port 80 which overshadowed the C&C traffic, we could not detect C&C traffic. Therefore, we had to reduce the amount of HTTP traffic to be analyzed. This enabled the detection of botnet C&C traffic in HTTP traffic as shown in Figs. 8 and 9. The period of the bot in HTTP traffic on port 80 was 31 min.

The results show that the address count sequences are more robust than the packet count sequences and they can resist the presence of background traffic. This is because there are less fluctuations in address count sequence due to the few number of distinct addresses (i.e., hosts) compared to the number of packets that would be sent and received from these hosts.

We also reported the effect of the network's firewall on the duty cycle of the packet and address count sequences. The periodic behavior in our study had a low duty cycle. This was because the firewall blocks inbound traffic on suspicious ports, which in our analysis was sent in response to the outbound botnet C&C traffic.

Acknowledgments This Project was funded by the National Plan for Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia, Award Number 10-INF1279-02.

References

1. , Porras P, Stoll J, Lee W (2009) Active botnet probing to identify obscure command and control channels. In annual computer security applications conference (ACSAC '09). Honolulu.Gu G, Yegneswaran V, Porras P, Stoll J, Lee W (2009) Active botnet probing to identify obscure command and control channels. In annual computer security applications conference (ACSAC '09). Honolulu
2. Silva S, Silva R, Pinto R, Salles R (2013) Botnets: a survey. Comput Netw 57
3. Demarest J (2014) Taking down botnets: public and private efforts to disrupt and dismantle cybercriminal networks. In U.S. senate, committee on the judiciary, subcommittee on crime and terrorism. Washington
4. FBI (2013) FBI Statement on Botnet Operation Available: http://www.fbi.gov/news/news_blog/botnets-101/fbi-statement-on-botnet-operation
5. SOPHOS (2014) Security threat report 2014. Smarter, Shadier, Stealthier Malware
6. Ha D, Yan G, Eidenbenz S, Ngo H (2009) On the effectiveness of structural detection and defense against P2P-based botnets. In IEEE/IFIP international conference on dependable systems & networks (DSN). Lisbon
7. Zeidanloo HR, Zadeh MJ, Safari M, Zamani M (2010) A taxonomy of botnet detection techniques. In 3rd IEEE international conference on computer science and information technology (ICCSIT). Chengdu
8. Tao C, Futai Z (2012) Detecting HTTP botnet with clustering network traffic. In 8th international conference on wireless communications, networking and mobile computing (WiCOM), 2012. Shanghai
9. Singh N (2015) IRC botnets alive. Evolving, Effective &
10. Vijayan J (2015) IRC botnets are not quite dead yet
11. Eslahi M, Salleh R, Anuar NB (2012) Bots and botnets: an overview of characteristics, detection and challenges. In IEEE

international conference on control system. Computing and Engineering, Penang
12. Zhuge J, Han X, Guo J, Zou W, Holz T, Zhou Y (2007) Characterizing the IRC-based botnet phenomenon. China Honeynet Technical Report
13. Rodríguez-Gómez R, Maciá-Fernández G, García-Teodoro P, Steiner M, Balzarotti D (2014) Resource monitoring for the detection of parasite P2P botnets. Comput Netw 70
14. Garg S, Peddoju SK, Sarje AK (2016) Scalable P2P bot detection system based on network data stream. Peer-to-Peer Networking and Applications 9:1209–1225
15. Jiang H, Shao X (2014) Detecting P2P botnets by discovering flow dependency in C&C traffic. Peer-to-Peer Networking and Applications 7:320–331
16. Choi H, Lee H (2012) Identifying botnets by capturing group activities in DNS traffic. Comput Netw 56
17. Schiller CA, Binkley J, Harley D, Evron G, Bradley T, Willems C, Cross M (2007) Botnets: the killer web app: Andrew Williams
18. Gu G, Zhang J, Lee W (2008) BotSniffer: detecting botnet command and control channels in network traffic. In the 15th network and distributed system security symposium (NDSS'08). San Diego
19. Jackson AW, Lapsley D, Jones C, Zatko M, Golubitsky C, Strayer WT (2009) SLINGbot: a system for live investigation of next generation botnets," in cybersecurity application and technologies conference for homeland security (CATCH). Washington
20. Lippmann R, Haines J, Fried D, Korba J, Das K (2000) The 1999 DARPA off-line intrusion detection evaluation. In 3rd international workshop on recent advances in intrusion detection (RAID). New York
21. LBNL/ICSI (2013) LBNL/ICSI Enterprise Tracing Project Available: http://www.icir.org/enterprise-tracing
22. Tarng W, Den L, Ou K, Chen M (2011) The analysis and identification of P2P Botnet's traffic flows. Int J Commun Netw Inf Secur 3
23. X. Yu, X. Dong, G. Yu, Y. Qin, D. Yue, and Y. Zhao, "Online Botnet Detection Based on Incremental Discrete Fourier Transform," Journal of Networks, vol. 5, May, 2010
24. Arshad S, Abbaspour M, Kharrazi M (2011) Sanatkar H. An Anomaly-Based Botnet Detection Approach for Identifying Stealthy Botnets, In Computer Applications and Industrial Electronics (ICCAIE)
25. CISCO (2011) NetFlow Version 9 Flow-Record Format Available: http://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9.html
26. Ma X, Guan X, Tao J, Zheng Q, Guo Y, Liu L, Zhao S (2010) A novel IRC botnet detection method based on packet size sequence. In IEEE International Conference Communications (ICC), Cape Town
27. AsSadhan B, Moura JMF (2014) An efficient method to detect periodic behavior in botnet traffic by analyzing control plane traffic. J Adv Res 5(4):435–448
28. Goebel J, Holz T (2007) Rishi: identify Dotcontaminated hosts by IRC nickname evaluation. In First Workshop on Hot Topics in Understanding Botnets, Cambridge
29. Oppenheim AV, Schafer RW (2009) Discrete Time Signal Processing, 3rd edn edn Pearson
30. Stoica P (2005) Spectral analysis of signals. Randolph L. Moses, Ohio State University. Prentice hall.
31. AsSadhan B (2009) Network traffic analysis through statistical signal processing methods. Carnegie Mellon Univ
32. Endace (2012) DAG 7.5G2 Card User Guide, Version
33. Endace (2009) Network tapping technical overview, Version
34. Pei Z, Xiao-hong H, Min-qi L, Chun-yu N, Yan M (2010) Fast restorable prefix-preserving IP address anonymization for IPv4/IPv6. J China Univ Posts Telecommun 17
35. Fan J, Xu J, Ammar MH, Moon SB (2004) Prefix-preserving IP address anonymization measurement-based security evaluation and a new cryptography-based scheme. Int J Comput Telecommun Netw46.
36. Kumar SA (2012) Conficker botnet prevention: crypto-pan algorithm. Int J Eng Sci 1
37. Lapworth L (2013) The Perl Programming Language Available: http://www.perl.org

Basil AsSadhan is an assistant professor at the Electrical Engineering Department at King Saud University. He received his Ph.D. and MS degrees in Electrical and Computer Engineering from Carnegie Mellon University, and the University of Wisconsin, respectively. His research interests are in the areas of cybersecurity, network security, and network traffic analysis and anomaly detection.

Abdulmuneem Bashaiwth is a Ph.D. Student at the Electrical Engineering Department at King Saud University. He received M.Sc. degree in Electrical Engineering from King Saud University in 2015. He Received a B.Sc. degree in Electrical and Electronic Engineering from Hadhramout University in 2005. His research interest is in the area network traffic analysis and anomaly detection.

Jalal Al-Muhtadi is the Director of the Center of Excellence in Information Assurance (CoEIA) at King Saud University. He is also an assistant professor at the Computer Science Department at King Saud University. He received his PhD and MS degrees from the University of Illinois at Urbana-Champaign, USA. He has over 40 scientific publications in the areas of cybersecurity, information assurance and Internet of Things.

Saleh Alshebeili is professor and chairman (2001–2005) of the Electrical Engineering Department, King Saud University. He has more than 25 years of teaching and research experience in the area of communications and signal processing. Dr. Alshebeili is member of the board of directors of King Abdullah Institute for Research and Consulting Studies (KAI-RCS, 2007–2009), member of the board of directors of Prince Sultan Advanced Technologies Research Institute (PSATRI, 2008–2017), the managing director of PSATRI (2008–2011), the director of Saudi-Telecom Research Chair (2008–2012), and co-founder and director (2011-Present) of the Technology Innovation Center, RF and Photonics for the e-Society (RFTONICS).