Frequency Domain Analysis of Large-Scale Proxy Logs for Botnet Traffic Detection

Giovanni Bottazzi
Department of Civil Engineering and Computer Science
University of Rome – "Tor Vergata"
Via del Politecnico, 1 – 00133 Rome
gbottazzi73@gmail.com

Giuseppe F. Italiano
Department of Civil Engineering and Computer Science
University of Rome – "Tor Vergata"
Via del Politecnico, 1 – 00133 Rome
giuseppe.italiano@uniroma2.it

Giuseppe G. Rutigliano
Department of Electronics Engineering
University of Rome – "Tor Vergata"
Via del Politecnico, 1 – 00133 Rome
rutigliano@ing.uniroma2.it

ABSTRACT

Botnets have become one of the most significant cyber threats over the last decade. The diffusion of the "Internet of Things" and its for-profit exploitation, contributed to botnets spread and sophistication, thus providing real, efficient and profitable criminal cyber-services. Recent research on botnet detection focuses on traffic pattern-based detection, and on analyzing the network traffic generated by the infected hosts, in order to find behavioral patterns independent from the specific payloads, architectures and protocols. In this paper we address the periodic behavioral patterns of infected hosts communicating with their Command-and-Control servers. The main novelty introduced is related to the traffic analysis in the frequency domain without using the well-known Fast Fourier Transform. Moreover, the mentioned analysis is performed through the exploitation of the proxy logs, easily deployable on almost every real-world scenario, from enterprise networks to mobile devices.

CCS Concepts

• Security and privacy → Intrusion/anomaly detection and malware mitigation → Malware and its mitigation.

Keywords

proxy, mining, logs, botnet, frequency domain

1. INTRODUCTION

One of the most insidious threats in the cyber domain is currently represented by the diffusion of botnets, which are networks of infected computers (called bots or zombies) completely managed by the botmaster, controlling the activities of the entire network and giving orders to every single zombie through some Command-and-Control servers (C2). Botnets are typically propagated through malware. However, due to the evolution of both the exploits of new vulnerabilities and the attack techniques that have not yet been identified as such, botnets cannot be easily detected by traditional security solutions such as firewalls, IDS/IPS and antiviruses.

SIN, July 20-22, 2016, ,

DOI: http://dx.doi.org/10.1145/2947626.2947634

Among the approaches in botnet detection outlined over the years, we consider, in this paper, the analysis of traffic generated by bots communicating with their C2 servers (post-infection stage), following a common pattern of interactions between clients (zombies) and servers, essential for bot synchronization, command deployment and for storing exfiltrated data.

The communication between zombies and C2, always started by bots, outlines very often characteristics of periodicity and timeliness [1], see, e.g., the Citadel configuration file in [2], and allows the botmasters to get a "complete picture of the situation" almost in real time and plan the subsequent actions to be commissioned to the bots.

The periodical communication also appears to be particularly interesting because it is completely independent from the agents, the protocols, C2 addressing (hard-coded IP addresses, DGA, Fast Flux, etc.) and from encryption techniques, simply because it takes into consideration only the timestamp of connections, but used in the frequency domain.

In the following paragraphs, we will describe the two contributions we believe to be innovative, to the best of our knowledge, related to how the periodic actions made by an agent can be efficiently highlighted in the frequency domain. We propose a method that does not use the well-known Fast Fourier Transform (FFT), for the application of which are needed a number of interventions closely related to the best-fitting of a function of continuous signals to a bursty dataset.

For example, the efficient application of the FFT requires an amount of samples that must be a power of two, the signal must be analyzed during all the time interval used for the observation and the signal must be properly described among two positive observations – when we have proxy connections – creating "acceptable" rising and falling edges – where we have no connections. Moreover, our proxy-based implementation can be easily deployed on any real-world scenario. In order to confirm our experiments, we developed an ad-hoc agent able to contact an Internet domain with either a fixed timing and a random timing within a fixed range.

Finally, our method makes it possible to find what are the specific connections outlining a periodic behavior, and thus suspicious, in completely "blind conditions", without any previously acquired knowledge about the bots' activity.

2. RELATED WORK

The frequency domain analysis is widely used in signal processing. The continuous signals are in fact representable as a sum of "simple" sinusoidal signals of different frequencies. The set

– and the amount – of all the frequencies contained in a signal is defined as the frequency spectrum of the signal. The frequency domain analysis allows to point out characteristics not easily observable through the time domain analysis, especially in complex situations in which several signals overlap.

For the above reasons, recently, many researchers proposed botnet detection methods, feeling that the pre-programmed botnet activities related to C2 traffic, could highlight spatial-temporal correlations and similarities [1]. Exploiting these similarities, could result in a framework whose operation is independent from protocols, architectures and payloads used, just because it can exploit correlated communication flows [3].

These insights allowed to observe that the communication flows between bots and C2 servers, disclose regularities detectable by means of the FFT, widely used in signal theory [4]. In fact, in [5], three main features were used as the basis for the extraction of network traces (to be understood as a sequence of chronologically-ordered flows between two network endpoints):
- the average time interval between the start times of two subsequent flows in the trace;
- the average duration of connections;
- the average number of source bytes and destination bytes per flow.

Other approaches [6, 7, 8] exploited the frequency analysis of suspicious flows, based on tuples of information, composed again by source and destination addresses, bytes exchanged, average packet size, etc., whose main target is to evaluate the effectiveness of a specific methodology tested in a controlled environment, but especially with a limited number of samples collected in a restricted observation period. The last two papers we believe important to mention are [9, 10], whose main novelty is the use of frequency analysis based on FFT, but applied to DNS queries. All the aforementioned approaches base their results on the use of the FFT applied to a certain type of network flows, often extracted using NetFlow, or to a set of DNS queries.

Unfortunately, all the data used so far (log files, pcap files, etc.) reflect the usual operation of computer networks, characterized by bursty traffic and inactivity periods, that is quite different from the definition of a continuous time-variant system. The bursty traffic involves the need to handle large amounts of data many of which, especially for the inactivity periods, do not provide useful information to the analysis. Therefore, there is the need to pre-process the data presenting impulsive behaviors, in order to make them more similar to a continuous function and thus to prevent the frequency scattering.

Finally, starting with a large number of samples in the time domain, the FFT calculates, again, a large number of samples in the frequency domain, and this makes it more difficult to analyze the frequency spectrum, both methodologically and computationally, looking for anomalies.

3. PROPOSED METHOD

We propose a method that does not use the Fourier Transform. We prefer, instead, an approach based on the underlying principles of the histograms applied directly to the data to be examined, so as to minimize the requirements in terms of memory and computational cost.

The main idea is to properly mine the proxy logs for extracting knowledge about the periodic behavior of bots, concealed behind the connection timing. Hence, each connection trace stored in the proxy logs, referred in the following as hit, was used to compute the time difference existing between two subsequent hits (in case of logs related to multiple workstations the first hit of every source IP address has been set to zero).

The "differential" values thus obtained, referred in the following as periods, were then clustered in order to find anomalous peaks compared to the shape of the resulting distributions, a.k.a. periodograms, for every single workstation (after clustering). Each anomalous peak denotes, as a result, some periodical agent's activity.

Of course, the visual analysis of anomalous peaks can be used only to analyze the periodograms of single workstations, while for multiple workstations must be used a method, similarly to the signal theory, as better described in Section 5, able to separate signal from noise, that in our case are respectively the traffic generated by bots and by humans.

We stress that our method is able to find the periodical actions of bots, similarly to what has been already done by previous approaches, but offers additional advantages since:
- it does not require any data preprocessing (flows extraction, average packet length, average duration of the connections, etc.);
- it does not need any previously acquired knowledge on bot's behavior;
- it is completely independent from the number of periods composing the observation, while the FFT treats the whole frequency spectrum.

Moreover, the exploitation of the method on proxy logs allowed us to build a bot detection tool that can be implemented efficiently and effectively in almost every practical scenario.

4. EXPERIMENTAL SETUP

We considered the proxy logs of a wide public corporate network over the whole day of 25th January 2016. The network consists of more than 60,000 workstations, distributed all over Italy and used by more than 100,000 users.

This corporate network deals with critical information ad it is thus equipped with its own security organization and infrastructure, which is composed by managed perimeter security systems and endpoint protection systems. Moreover, the current corporate policies forbid the interaction between endpoints and the Internet with protocols other than http and https.

The logs were stored in huge text files in W3C format (more than 100 GBytes of daily logs for the entire network). We considered only four log file fields, which contain information about the client IP address contacting the Internet, the IP address to be contacted, the URL involved in the connection and when the connection occurred (time).

In order to properly test our method, we developed and injected within the network infrastructure, an ad-hoc agent able to contact two different Internet URLs with either a fixed timing and a random timing within a fixed range, trying to mimic the behavior of a real malicious agent.

5. LOG ANALYSIS AND RESULTS

The first issue to manage and solve in treating large text files is purely of logistic nature. To mine log files, we used the free tool Log Parser [11] to implement a script formatted in a SQL-like language, in order to extract the data stored in the W3C files and upload the results to a data base.

With this method it was possible to store, effectively and efficiently (e.g., in few minutes using a common desktop machine), all the data associated with our experiment, extracting first the logs related to one workstation, from about 100 GBytes of textual logs, and then expanding the experiment to a wider set of workstations. It is important to note that the same log extraction method can be easily scaled to bigger size logs as already demonstrated in a previous work [12] where, although in a very different context, a starting dataset of almost 3 TBytes has been used.

We made our first experiment injecting our ad-hoc agent capable of making 75 hits towards an Internet URL with a fixed timing of 30 seconds (32 minutes in total). Once the data have been extracted and uploaded to the data base, we grouped the periods, as described in Section 3, for just one workstation. Fig. 1a plots the resulting periodogram and shows how many times (y-axis) the workstation has actually had a period of t seconds of inactivity (x-axis). The initial data set of more than 11,000 hits, for only one host, was reduced to 250 clusters each containing the number of hits started t seconds after the previous one. The total number of clusters does not cover all the possible periods (in seconds) we can have in a day. The nature of proxy logs is such that we can have periods with high network activity and periods with no activity (those we want to exploit for finding periodical behaviors).

As we can see, most of the occurrences tend to collapse in the small periodicities of the series (left part of fig. 1a) denoting the typical human usage of the Internet. We deleted the first four occurrences of the continued line series, related to periods of 0, 1, 2 and 3 seconds, in order to graphically highlight the lowest values. When someone (human) activates a hyperlink, by clicking or typing directly the URL, a series of events almost concurrent are triggered (all the simultaneous events, seen in the frequency domain, fall within the same period, zero seconds).

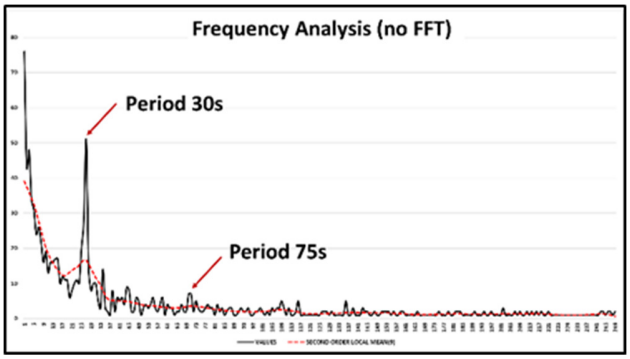

Fig. 1a. Frequency analysis computed by clustering the differential timing between proxy logs with the injection of an agent acting in the fixed period of 30s.

Thus, the time spent by the user (human) to read the content of an Internet page, causes the workstation inactivity (the reason why we observed low values in the large periodicities on the right part of fig. 1a). Bots (or agents) instead act, by definition, with time periodicities and the effectiveness of this method makes it possible to "isolate" these periodicities because human behavior has numerous periods of inactivity where can fall many of the actions of an agent. In fact, we may notice that the 27th element of the series in fig. 1a (51 hits, period 30 seconds) denotes an anomalous peak compared to the shape of the distribution, related to the activity of the agent we introduced for testing within the network. It is interesting to note that, looking at the peak with a period of 30s, some hits were partially "scattered" because 51 hits out of 75, made by the agent, have been logged in the right period, 11 of which are not related to the agent's activity. This mean that only the 53% of the logged hits started during an inactivity period of the workstation, and so they were logged correctly within the 30s-period, while the others are "hidden" over other periods. The test described in fig. 1a allowed us to find the presence of another agent, different from the one we injected for testing. In fact, the 71st element of the series (7 hits, period 75 seconds), is related to the URL "nexus.officeapps.live.com", the service used by some Microsoft software as Office 365, to report periodically the software health.

As we can see in fig. 1b the same data treated with the FFT resulted in a much noisier distribution. This is mainly due to – as described in Section 1 – the high number of frequency samples (as many as those in the time domain), although we:

- built a timeline able to cover all the observation periods, whose elements is a power of two;
- created "acceptable" rising and falling edges, related to periods of inactivity.

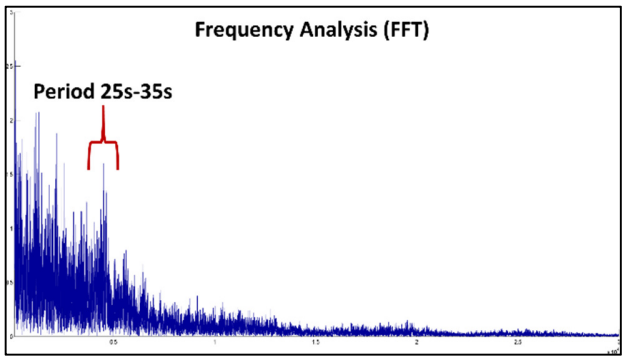

Fig. 1b. Frequency analysis computed with FFT.

In the second experiment, we injected our ad-hoc agent whose actions were scheduled in random periods within a small range, chosen in this case within a range of 10 seconds (from 25 to 35 seconds), in order to simulate a possible evasion technique made through the random distribution of the agent's action over different periods. Again, we can highlight anomalous peaks, compared to the shape of the distribution (fig. 2), but with a more marked largeness, because of the distribution of periodicity over a range period of 10 seconds.

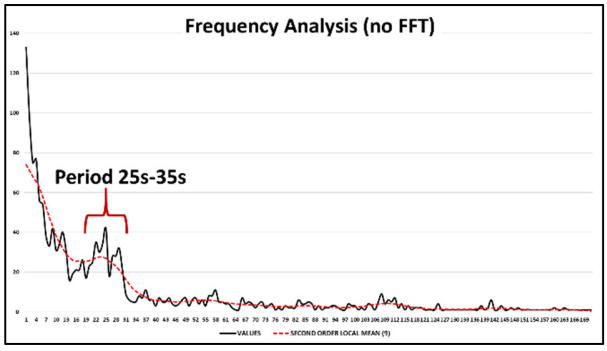

Fig. 2. Frequency analysis computed by clustering the differential timing between proxy logs with the injection of an agent acting in randomized periods within the range of 25s and 35s.

In order to increase the randomness of the agent, both with fixed and randomized periods, we made a third experiment,

deleting some agent's logs, depicting a behavior that skips randomly some hits. The results are plotted in fig. 3, where the agent's actions can be still isolated, partially with its "**original behavior**" (periods of 30 seconds and period range 25-35 seconds) and partially with its "**harmonic behavior**" (period 60 seconds and period range 50-70 seconds). The experiment just described reveals that the periodicity of actions made by an agent is a very strong feature able to "resist" also to a second order of randomness.

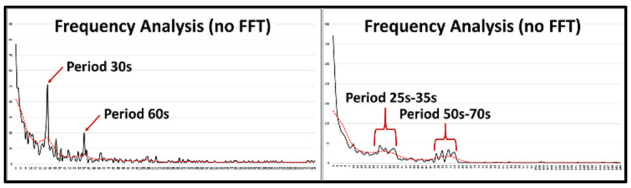

Fig. 3. Fixed period and random period within a small range, with harmonic behavior.

Hence, the "success" factor, from the botmaster's point of view, is the number of "useful hits" during the observation period, to be understood as the connections whose periodicity is obfuscated by normal traffic that, if reduced in amount, can be completely "hidden" within other periods.

Thus, the method we implemented, exploits the agent's hits, whose periodical behavior fits completely within the inactivity periods of the workstation. For this reason, implementing an agent with a high inactivity time, on one hand increases the probability of being detected, because its actions fall within an area where there are few hits (little activity), but on the other hand it increases the probability that its extended periodical behavior is "obfuscated" by other traffic, even other agent's traffic with a shorter periodicity, making it invisible.

Stressing the concept of extended periodicity, we made another experiment, injecting our ad-hoc agent with a periodicity of 30 minutes, making 15 hits in total. As expected, the agent's actions were completely concealed by shorter periods. In fact, as you can see in the left side of fig. 4, no peaks are related to a period of 30 minutes. Instead, removing from the proxy logs the hits related to just two well-known licit Internet domains (those related to the operating system and the antivirus), we had a great enhancement of agent's visibility (right side of fig. 4).

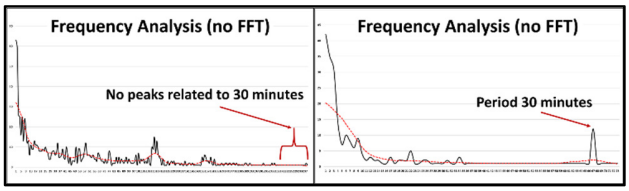

Fig. 4. Trace of agent's activity with a periodicity of 30 minutes.

Hence, to the best of our knowledge, both attackers and defenders have to balance the scalability and agility of the botnet to deploy/detect, with the increasing risk of detection associated with an increasing number of C2 server connections.

In order to automatically identify the peaks contained within the periodograms, similarly to the signal theory, we need a method, able to separate signal from noise, that in our case are the continued line series (signal – the traffic generated by bots) and the dotted line series (noise – the traffic generated by humans). The dotted line series, the noise, has been considered to be as the local average of the second order (the average of the average), computed on a neighborhood of 9 values of the continued line series, the signal (1).

The local average of the first order is the one shown in (2). As you can see, we excluded the value of x_i in the first order average, $Avg_9(x_i)$, in order to better highlight anomalous peaks.

$$\forall\, i, Noise(x_i) = \frac{\sum_{j=i-4}^{j=i+4} Avg_9(x_j)}{9} \quad (1)$$

$$\forall\, i, Avg_9(x_i) = \frac{\sum_{j=i-4}^{j=i+4} x_j}{8}\ i \neq j \quad (2)$$

Finally, we considered as anomalous peaks (Signal to Noise Ratio), all the signal values greater than 1.5 times the noise values (3).

$$\forall\, i, SNR(x_i) = \begin{cases} Peak, & x_i \geq 1.5\ Noise(x_i) \\ NoPeak, & otherwise \end{cases} \quad (3)$$

In order to complete our experiments, we applied the method to a wider set of workstations (using the proxy logs of the hosts of a /16 network). With the aforementioned SNR method, it was possible to highlight, for each source IP address, what are the suspicious periods, because deriving from the action of a bot. Tracing back the specific URLs, inclusive of the source and destination IP addresses causing the frequency peaks, is almost instantaneous. The scenario, except for the number of logs treated, was always the same: we just computed and clustered, for each workstation involved, the timing differential values of the proxy hits.

In order to reduce the amount of data to be inspected and to increase the agents' action visibility, we deleted all the domains included in the ALEXA top 500 web sites, reducing the hits from more than 8 million rows to 4 million (about 1,200 distinct source IP addresses). The application of our method (differential values calculation e subsequent clustering), allowed us to reduce the data set to about 90,000 rows, storing the periodograms of all the source IP addresses inspected.

Further, using the SNR described in equation (3), we were able to further reduce the dataset to about 8,000 different peaks. Among the agent's activity peaks, we deleted all the hits related to common refreshing actions made by the web servers of public web mails, online newspapers, online weather forecasts etc., further reducing the dataset to a few hundreds of hits.

Hence, among the resulting dataset, considering that we worked within a network security environment highly managed and, again, without any previously acquired knowledge, we were finally able to find (extract in fig. 5) a number of so called Potentially Unwanted Programs (a.k.a. PUPs), that can do harmful things like wrapping malicious code [13] and the actions of a real malicious agent installed on a single workstation (fig. 6).

As we can see in fig. 6, we found a workstation highlighting a peak of just two hits (period of 1 minute and 21 seconds in just 3 hours of total activity), trying to contact both the URLs:

- "ww1.generaldownloader.com";
- "1.update.generaldownloader.com".

The aforementioned URLs have been recognized as participating in malicious activity, and are confirmed to be hosting exploit code, malware, trojans, spyware and other malicious tools, included the famous ZeroAccess botnet traffic [14]. Of course we didn't consider, again, the peaks related to periods of 0 and 1 seconds.

IP address	DOMAIN	SNR	PERIOD (mm:ss)	CATEGORY
213.254.17.159	c5.zedo.com	3.3	01:01	PUP
173.194.112.13	redirector.gvt1.com	2.9	05:00	PUP
144.76.116.147	p.lp4.io	3.61	05:01	PUP
74.115.0.211	a433.com	1.57	12:00	PUP
216.52.1.12	loadm.exelator.com	4.2	00:15	PUP
195.93.42.2	adserver.adtech.de	1.7	01:11	PUP
37.252.170.26	ib.adnxs.com	1.6	00:20	PUP
54.208.250.38	q.adrta.com	3.1	01:50	PUP
208.81.196.145	**ww1.generaldownloader.com**	**1.7**	**01:21**	**MAL**
94.229.72.117	**1.update.generaldownloader.com**	**1.7**	**01:21**	**MAL**
193.45.15.70	193.45.15.70	8	01:01	PUP
198.41.190.38	rum-static.pingdom.net	1.8	01:01	PUP
50.31.164.173	bam.nr-data.net	7.7	04:00	PUP
37.114.77.112	www.sportatomic.com	2.5	01:20	PUP
69.16.175.10	notif.basememlog.com	2.2	06:00	PUP

Fig. 5. Extract of Peaks List.

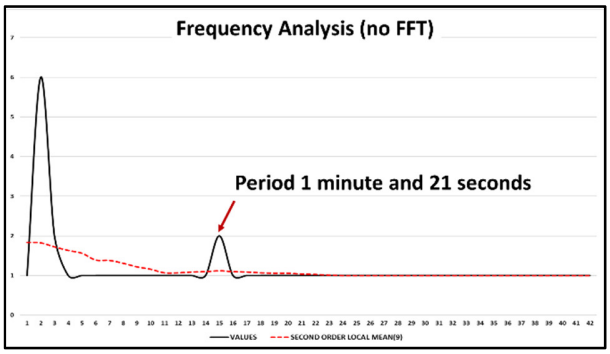

Fig. 6. Malicious agent activity.

6. CONCLUSIONS AND FUTURE WORK

Once the URLs have been extracted, the application of some white lists is of fundamental importance both for reducing the amount of data to be inspected and for deleting all the occurrences (periodical or not) related to known licit domains, in order to isolate malicious agents almost as if they were acting alone (without background noise). In this way, while introducing an overhead, it is possible to overcome the limit posed by all the frequency analysis techniques. As mentioned earlier, the hits contained in every single peak may have occurrences that do not concern any agent, but the great majority must necessarily refer to a periodic action. Furthermore, the frequency analysis described in this paper is particularly effective, given its absolute independence from the agent, and efficient, because it can fulfill the frequency analysis of the proxy logs exploiting only the observed traffic. Exploiting the proxy logs, however, has a high added value, given that they:

- are natively interposed to communications between client and server and, thus, can be used both to identify a suspicious Internet domain or to block the connections to it;
- can be deployed also in consumer environments such as laptops and mobile [15].

Hence, the methodology just described allowed us "to put in place" a highly scalable detection tool in almost every real-world scenario (the computational cost is proportional to the amount of data to be processed). For the future, we plan to extend the experiment on a much wider set of workstations and to test the tool on real malware family samples.

7. REFERENCES

[1] G. Gu, J. Zhang, and W. Lee, “Botsniffer: Detecting botnet command and control channels in network traffic”, in NDSS, 2008.

[2] Aditya K. Sood, Rohit Bansal, “Prosecting the Citadel botnet - revealing the dominance of the Zeus descendent”, Kaspersky Virus Bulletin, September 2014;

[3] G. Gu et al., “Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection”, in USENIX Security Symposium, 2008, pp. 139–154.

[4] AsSadhan B. et al. "Detecting botnets using command and control traffic", in Network Computing and Applications, 2009. NCA 2009. 8th IEEE International Symposium on. IEEE, 2009.

[5] F. Tegeler et al., “BotFinder: finding bots in network traffic without deep packet inspection.”, In Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '12), 2012.

[6] P. Tuhin, et al. "Fast-flux botnet detection from network traffic." India Conference (INDICON), 2014 Annual IEEE. IEEE, 2014.

[7] Kinjal S. Thaker, “Modelling and Detection of Camouflaging Worm at an Advance Level”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 10, October-2015, pp. 758-762.

[8] Soniya Balram and M. Wilscy, “User Traffic Profile for Traffic Reduction and Effective Bot C&C Detection”, International Journal of Network Security, Vol.16, No.1, Jan. 2014, pp.46-52.

[9] J. Kwon et al., “PsyBoG: Power spectral density analysis for detecting botnet groups”, in Proceedings of the 9th IEEE International Conference on Malicious and Unwanted Software, MALCON 2014.

[10] J. Kwon et al., “PsyBoG: A scalable botnet detection method for large-scale DNS traffic”, in Computer Networks 97 (2016), pp. 48–73.

[11] G. Giuseppini, M. Burnett, J. Faircloth, D. Kleiman, “Microsoft Log Parser Toolkit: A complete toolkit for Microsoft's undocumented log analysis tool”, ISBN-13: 978-1932266528.

[12] G. Bottazzi, G. F. Italiano, “Fast Mining of Large-Scale Logs for Botnet Detection: A Field Study”, in Proceedings of the 3rd IEEE International Workshop on Cybercrimes and Emerging Web Environments, Liverpool, UK, October 2015.

[13] Kaspersky Security Bulletin 2014. Overall Statistics for 2014.

[14] OTX – Alien Vault. https://otx.alienvault.com/indicator/ip/208.91.196.145/

[15] G. Bottazzi et al., “MP-Shield: A Framework for Phishing Detection in Mobile Devices”, in Proceedings of the 3rd IEEE International Workshop on Cybercrimes and Emerging Web Environments, Liverpool, UK, October 2015.