Survey On Botnet Detection Techniques

Dr. Pervaiz khan Rabbani And Umair Mujtaba

Abstract: Internet security problems remain a major challenge with many security concerns such as Internet worms, spam, and phishing attacks. Botnet has recently been recognized as one of the most significant security threats/worms of the Internet. Botnet attacks degrade the status of Internet security. Although research on the topic of botnets is relatively new, it has been the subject of increasing interest in recent years and has spawned a growing number of publications. This paper presents a comprehensive review of the latest techniques for botnet detection, in addition to figuring out the trends of previous and current research. In this paper we discuss some of the botnet detection techniques and compare their advantages, disadvantages and features used in each technique.

[Pervaiz khan Rabbani AND Umair Mujtaba. **Survey On Botnet Detection Techniques.** *Researcher* 2016;8(9):56-62]. ISSN 1553-9865 (print); ISSN 2163-8950 (online). http://www.sciencepub.net/researcher. 6. doi:10.7537/marsrsj080916.06.

Keywords: Survey; Botnet; Detection; Technique

1. Introduction

Now a days one of the most significant threats to the Internet is the threat of botnets, which are networks of compromised machines under the control of an attacker. It is difficult to measure the extent of damage caused on the Internet by botnets, but it is widely accepted that the damage done is significant. Therefore a number of ad hoc methods exist to detect and stop botnets, and these methods continue to mature. As techniques for botnet detection and mitigation improves, the robustness and resiliency of botnets will also advance.

Today, the most easily detected botnets use IRC as a form of communication for command and control (C&C). IRC has many properties that make it attractive for an attacker such as its redundancy, scalability, and versatility. Further, there is a large base of knowledge and source code for developing IRC-based bots. Many botnet authors reuse existing code in order to create their own botnet.

One of architectures that are used for botnet communication is peer-to-peer. In a peer-to-peer architecture, there is no centralized point for C&C. Nodes in a peerto-peer network act as both clients and servers such that there is no centralized coordination point that can be incapacitated. If nodes in the network are taken offline, the gaps in the network are closed and the network continues to operate under the control of the attacker.

Once the network infected with a bot, the victim host will join a botnet, which is a network of compromised machines of a malicious entity, typically referred to as the botmaster. They use different techniques for evasion from user and detection like using multiprocess bot instead of single process bot. [1] Botnets are the primary means for cyber-criminals to carry out their nefarious tasks, such as sending spam mails, launching denial-of-service attacks, or stealing personal data such as mail accounts or bank credentials. This reflects the shift from an environment in which malware was developed for fun, to the current situation, where malware is spread for financial profit. Denial-of-service (DoS) attacks, phishing, spamming, key logging, click fraud, identity theft and information exhilaration here is main hazardous behavior which is associated with the botnet. Botnets apply a self-propagating function to infected hosts. Given the importance of the problem, significant research effort has been invested, to gain a better understanding of the botnet phenomenon.

Another approach to study botnets is to perform passive analysis of secondary effects that are caused by the activity of compromised machines [2]. For example, researchers have collected spam mails that were likely sent by bots. These models collecting the data through monitoring activities which can be tracked without interfering with the environment or tampering with the evidence. Some researchers analyzed IRC traffic, capable of identifying botnet related activities. A more active approach to study botnets is via infiltration. It contains approaches that involve interaction with the information sources being monitored. Infiltration of botnets can be divided into: software and hardware based techniques. The first covers research on the bot executable and monitored traffic to achieve control and conduct measurements. They can be applied if access to the command-and control server is possible and may be used to wiretap the communication. According to the command and control (C&C) models, botnets are separated into two groups, centralized (e.g., IRC and HTTP) and distributed (e.g., P2P). Centralized botnet utilize two mechanisms to get the command from the server, which is push and pull. In the push system, bots are associated to the C&C server (e.g., IRC server) and wait for the commands from the botmaster. In contrast presented in the pull mechanism, the botmaster sets the commands in a file at C&C server (e.g., HTTP server), and the bot often connect to the server to read the most recent commands. While in centralized structure all bots receive the commands from a definite server, in distributed structure the command files will be mutual over P2P networks by botmaster and bots can use explicit search keys to find the available command files [3].

Botnet Analysis is to determine the path from a victim network or system through any intermediate systems and communication pathways, back to the point of attack.

Static Analysis is also known as White box testing. It is the process of understanding the behavior of a program without executing it. The analysis checks the presence of viruses in file system such as firewall logs.

Dynamic Analysis called as Black box testing differs from static analysis is that the bot is executed, usually in a controlled environment.

How, what and where is done by bots, that is botnet forensics Analysis. Forensic is a discipline based on science & technology to investigate and establish facts in criminal & civil courts. It deals with collecting, analyzing and helps in presenting evidences in a court of law. Network forensics is the science that deals with capturing, recording, and analysis of network traffic for detecting intrusions and investigating them [4].

The rest of paper is organized as following. Section II introduces the botnet life cycle, its structure and architecture. Section III presents the techniques to mitigate the botnet detection and last in Section IV result and discussion.

2. Background and Motivation

There are three basic element of botnet which are bot, botmaster and c&c. These elements are used in survival of botnet. There are five phases involved in botnet life cycle. [5].

The *first* phase is the Initial Injection, where in a host is infected and becomes a potential bot. This phase is characterized by a regular computer infection procedure, which may be carried out in different ways as a typical virus infection would be, for instance, through unwanted downloads of malware from websites, infected files attached to email messages, infected removable disks, etc [6].

The *second* phase is the secondary injection which requires the first phase be successfully completed. In this phase, the infected host runs a program that searches for malware binaries in a given network database. When downloaded and executed, these binaries make the host behave as a real bot (or zombie). Downloading bot binaries is usually performed by FTP, HTTP or P2P protocols [6].

Third phase is scheduled every time the host is restarted to ensure the botmaster that the bot is taking element in the botnet and is capable to receive commands to perform malicious activities. After establishing the command and control channel the bot waits for commands to perform malicious activities.

Fourth phase is ready to perform an attack. Malicious/ Hazardous activities includes a wide range as information theft, performing DDoS attacks, extortion, monitoring network traffic, spreading malware, stealing computer resources, and unprotected computers, identity theft, phishing, spamming, manipulating games.

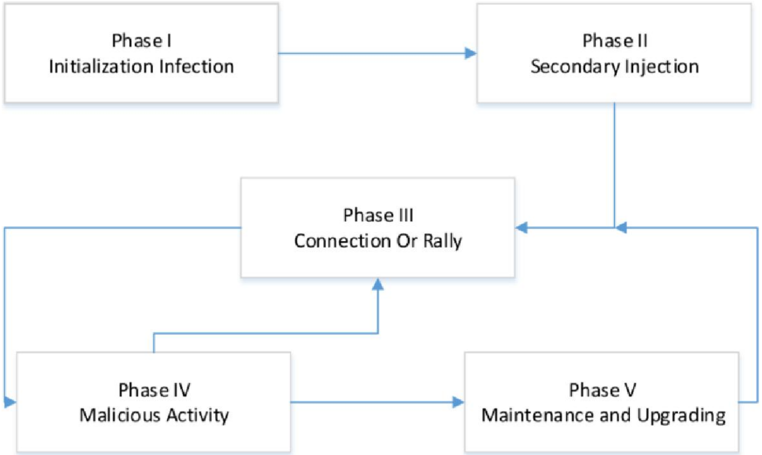

Figure 1. Botnet Life Cycle and surveys, etc.

Last phase is maintenance and up gradation which is the most important phase of botnet life cycle. Maintenance is necessary if the botmaster wants to keep his army of zombies. It may be necessary to update codes for many reasons, including evading detection techniques, adding new features or migrating to another C&C. This phase usually measures a susceptible step. As the botmaster intends to broadcast updates as soon as possible, some behavioral patterns of the stations belonging to the network may emerge and make the botnet detectable. Changes in behavior are typically observed, for instance, in DNS queries and file sharing, and among other areas. When bots are updated, they must establish new connections with the C&C infrastructure. [6]

2.1. **Botnet Architecture.** The strength of botnets lies in the potential of having a flexible network of connected computers which are controlled remotely. Therefore, different approaches are used to deal with the communication problems between the entities in the botnet. A number of architectures are proposed which include Internet Relay Chat (IRC) centralized architecture and decentralized P2P architecture which is recently extended to HTTP/S and Twitter based networks. [7]

2.1.1. *C&C Architecture.* The centralized C&C approach resembles the traditional client/server architecture. IRC protocol is an example of centralized C&C architecture wherein bots establish a strong communication channel between one or multiple connection points. Servers are deployed on the connection points wherein the responsibilities of sending commands to bots and delivering malware update takes place. IRC and Hyper-Text Transport protocol (HTTP) are considered as the main protocols in centralized architecture.

The advantages of centralized architecture include the following.

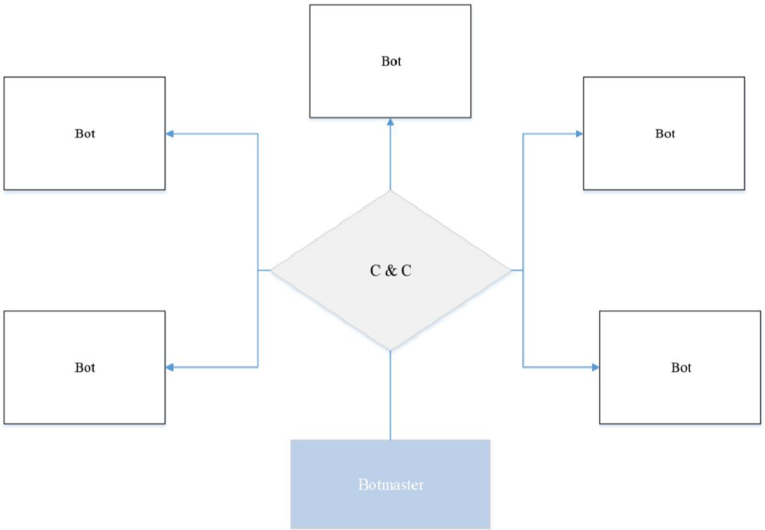

Figure 2. C & C Architecture

- This sculpt is easy to set up as it does not require any dedicated hardware.
- Quick Response time: because the server is directly coordinating with its bots without being intervened by a third party.
- Better coordination with the bot enemy.
- Easy accessibility: as there is direct coordination between the botmaster and their bots.
- Updates from the botmaster are effected timely.

According to C&C architecture is further classified into IRC and HTTP based approaches. The drawback of a centralized approach is that, the command and control (C&C) server is considered as a single point of failure [7], so it is quite easy to turn-off detected botnets.

2.1.2. *Decentralized Architecture.* Modern botnets require great flexibility and robustness to be able to handle large numbers of bots and to maximize profits. Botnets that have a decentralized architecture are more difficult to be mitigated because the discovery of several or even many bots does not necessarily mean the loss of the entire botnet because there is no central C&C server to be found and disabled.

A P2P botnet do not requires formal coordination and even if a node is taken offline by the defender, the network still remains under the control of the attacker. A botmaster transfer command to a bot peer, a command spreads all zombie peers by communicating with each other. They have the advantage of being more difficult to destabilize as they do not have a unique core which issues orders and/or sharing resources and information, making use of the facilities of traditional P2P networks allow a high connection and disconnection ratios. Each node has greater structural complexity because all of them can act as both, client and server, being more difficult to intercept and study. P2P botnets aim at removing the failure point which is the main limitation and vulnerability of centralized networks. P2P communication system is much strong, complex and does not guarantees message delivery or latency. Transferring command of P2P botnet is slow to compare with centralized botnet. This means that the compromise of a single bot does not necessarily mean the loss of the entire botnet.

Difference between Centralized and P2P botnet is following in table:

Table 1. Comparison Between Centralized and P2P

S.No	Parameters	Centralized	P2P
1	Tracking	Easier	Difficult
2	Single Point of failure	Can destroy the whole Botnet	Will not affect much
3	Cost incurred	Higher cost	Low
4	Risk of Hijacking	Hijacking of Bot controller can reveal the identity of Bot-master	Hijacking Bot peer cannot reveal the identity of Bot master
5	Command dis-tribution speed	Faster	Slower
6	Management	Easy	Difficult

2.1.3. *Hybrid Architecture.* A hybrid peer-to-peer botnet based on the unstructured P2P protocols. A hybrid botnet will be divided into servant and client bot. The servant bot receives the commands from the bot master, and forwards it to the client. The hybrid P2P botnet is equivalent to a C&C botnet where servant bots take the role of C&C servers, the number of C&C servers (servant bots) is greatly enlarged, and they interconnect with each other. In hybrid P2P Botnet, in comparison to current botnet, it is harder to shut down, monitored and hijacked. [8, 9]

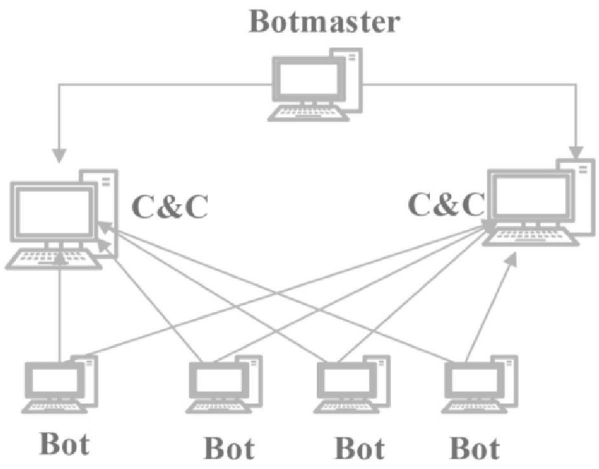

Figure 3. Hybrid Architecture

3. Botnet Detection Techniques

Botnet detection is perhaps one of the primary action that should be taken when discussing network security threats. Given the potential power of botnets to conduct different malicious activities and cyber warfare, detection techniques play an important role in this process. Researchers have developed several techniques for detecting such threats like *Botyacc* [10] and have proposed a number of botnet detection taxonomies.

We conclude that the comparison of different botnet detection methods with other proposals is highly beneficial for the botnet research community because it helps to objectively assess the methods and improve the techniques. Also, that the use of a good botnet dataset is paramount for the comparison. [11]

PeerShark is conversation based approach for P2P traffic which can differentiate P2P botnet traffic from real P2P traffic, and correctly categorize the exact P2P application running on a host inside a network. PeerShark significantly extends past works by addressing the challenging context of detection of stealthy P2P botnets in network traffic in the presence of benign P2P applications, and categorization of the specific type of P2P application running on a host. PeerShark has four modules(Packet Filtering Module, Conversation Creation Module, Conversation Aggregation Module and Classification Module). There four basic things(Duration of Conversation, Packets exchanged in conversation, Volume of Data exchanged in conversation and Median value of inter arrival time of packets in conversation) which can differentiate P2P botnets from P2P benign users. It aims to be P2P assistant to network administrators wanting to isolate P2P traffic and detect P2P botnets. A few limitations of PeerShark are fact that PeerSharks present approach gives abirds eye-viewof the conversations happening in the network. Being flow-oblivious many lower-level details (such as the Transport layer protocol) are neglected. If more than one P2P application is running between two peers (either benign or malicious), the flows from different applications are expected to get separated into different clusters (because of the different nature of flows seen in different application). [12] There is also another approach is purposed that detects the P2P botnet using signal processing. [13]

FPRF fuzzy pattern recognition-based filtering algorithm [14] based on common bot host behavior observed from DNS and TCP traffic. FPRF algorithm is divided into three stages: (1) traffic reduction: reduce input raw packet traces and speed up the processing of bots specific activities; (2) feature extraction: extract features from the reduced input packet traces; and (3) fuzzy pattern recognition: with extracted features, detect bot-relevant malicious domain names and IP addresses based on the maximum membership principle. FPRF has high detection rates of 95.29% and 95.24% for malicious domain names and malicious IP addresses, respectively. The experimental results based on normal traces also show a high traffic reduction rate of over 70% and low false positive rates (03.08%). Both results show that the FPRF algorithm is not only efficient but also highly accurate. In addition, the FPRF algorithm can detect inactive botnets, which can be used to identify potential vulnerable hosts.

Event Driven Log Analysis Software system that enables detection of botnet infection on the users system. The system architecture is consisted on three modules which are Main, Logparser and Analyzer. Main module uses other modules of system to analyse and provoid the results. The Logparser is created by Main module when network traffic is activated. The Logparser locate the firewall log data and parse the data in the format that is used by Analyzer module. The Analyzer search the defined tag and copies the information in a structure than can be easily searched and manipulated. The software system is designed to facilitate detection of botnets for users that lack knowledge about botnet analysis. Moreover the system notifies users, i.e. sends a warning message in case of their machines become a part of botnet. If the amount of outbound connections was too high, the user had to shut down all software programs that were using the network connection. Detecting botnets in such way has its flaws and perks. If outgoing network traffic is low but still in the range of the lowest threshold to issue a warning the user would get a warning but the computer could still be malware free just having some background process using the network.. The software program needs improvements, such as support for different log formats, support for other operating systems than Windows XP, and the botnet scan collecting and analysing functions. A scan function that will show the user their outgoing network traffic in real time is under development. [15]

BotGAD reveal both unknown domain names of C&C servers and IP addresses of hidden infected hosts. Using property group activity of botnet, BotGAD needs a small amount of data from DNS traffic, not all the traffic content or known signatures. BotGAD can detect botnets from a large-scale network in real-time even though the botnet performs encrypted communications. Moreover, BotGAD can detect not only individual botnets but also correlated evasive botnets. This method provides over 95% detection rates while generating less than 0.4% false positive rates and 5% false negative rates based on experiments with real-life campus and ISP DNS traces. It takes only a few minutes to analyze an hour-long DNS trace of a large ISP network. The evaluation results prove that BotGAD can automatically detect botnets in large scale networks. [16–18]

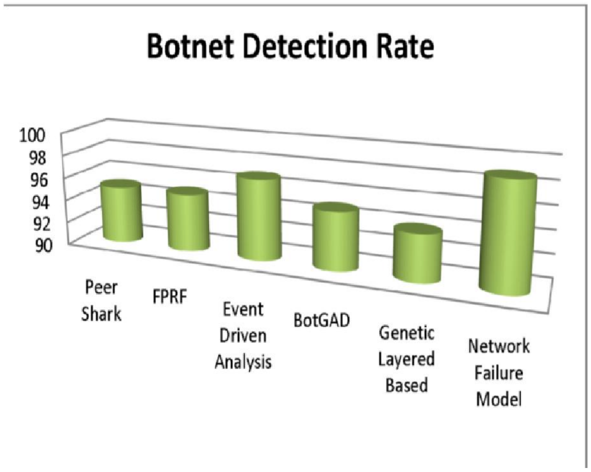

Figure 4. Botnet D etection Rate

Genetic Algorithm Based Layered System produces efficient detection which reduces false positive rate. Genetic Algorithm has four layers which are Layered HTTP botnet Detection, Packet Capturing Module, Genetic Algorithm for HTTP Botnet Detection and Detection Module. Packet capturing module intercepts or logs the traffic passing through the ports. Captured packets are analyzed to inspect the raw data. In Detection Module there are also used four layers(DDoS, Probe, R2L and U2R) to detect botnet. Genetic operation is calculated for each layer and if the numbers of packets are more than the genetic operation value, the corresponding attack is reported and the database is updated accordingly. In particular, it is found that such a system would be less computational intensive and more accurate.

Table 2. Botnet Detection Technique

Technique	Approach	Results	Disadvantages
Peer Shark	bots are detected by conversation among bot and botmaster	95%	Can't differentiate TCP and UDP packets
FPRF	Bot host behavior is observed and detected by DNS and TCP traffic	95 %	
Event Driven Analysis	bots are detected by activities which differ from human nature	97 %	
BotGAD	group activity of bots is observed and detected	95%	Botnets that do not use DNS are undetectable
Genetic Layered Based	Different Layers are used detect bots from traffic with with different prop-erties	94%	Only Http botnet are detected
Network Failure Model	Bots are detected by failure of command	99%	Botnet in which network failure never occured can't be detected

The probability of attack detection is calculated by following equation:

$F - value = 1/(.5)(1/precision + 1/Recall)$

Once a layer detects an attack, it is added to the Black/Gray list. Then there is no need for further layers to analysis that packet. Layered approach provides efficiency and reliability and the automation process of grey list and black list of the firewall provides robustness. It is further opined that rather than the active termination in GA, the cooperation of learned termination and enhanced convergence can lead to more optimized results. One of major drawback is it can only detect HTTP bots not other bots. [19,20]

Effective Bot Host Detection Based on Network Failure Models detects bot hosts based on their network failure models. This technique has two parts (Training Phase and Detection Phase). In the training phase, numerous benign traces, peer-topeer application traces, and bot traces, filter out non-failures, extract features from failure flows and build the classification model using the algorithm is collected. Detection Phase is similar to training phase ans classification is done using previous training. Bots generating network failures because of botnet-distributed design and implementation is intrinsic and inevitable. Evaluations show that the solution achieves a high detection rate (more than 99%) and low false positive rates (less than 0.5%).

The equations used in evaluation of performance are following:

(1) $precision = truepositives/(truepositives + falsepositives)$

(2) $recall = truepositives(truepositives + falsenegatives)$

(3) $F - measure = 2 * (precision * recall)/(precision + recall)$

(4) $FPrate = falsepositives/(truenegatives + falsepositives)$

Unlike other anomaly based approaches, the solution does not rely on aggregated group activities, does not need to examine payloads, and is able to detect bots in a short period. In addition to being efficient and robust, the proposed solution is lightweight in storage and computation costs. [21]

4. Conclusion

In this paper we have studied different architecture of botnets and also their detection method. We have analyzed that still there are many challanges in botnet detection. One of the most important is that there is no such a platform where the evaluation techniques are tested in real time. Botnet can spread all around the world so different network administrators should take action with cooperation.

The important part of future research is to make experiment prototype where the researcher can test their purposed detection technique regarding to botnet.

References

1. Y. Ji, Y. He, D. Zhu, Q. Li, and D. Guo, "A mulitiprocess mechanism of evading behaviorbased bot detection approaches," in *Information Security Practice and Experience*, vol. 8434. Springer International Publishing, 2014, pp. 75–89.
2. B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: Analysis of a botnet takeover," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*. ACM, 2009, pp. 635–647.
3. W. Lu, M. Tavallaee, G. Rammidi, and A. Ghorbani, "Botcop: An online botnet traffic classifier," in *Communication Networks and Services Research Conference, 2009. CNSR '09. Seventh Annual*, May 2009, pp. 70–77.

4. E. S. Pilli, R. Joshi, and R. Niyogi, "Network forensic frameworks: Survey and research challenges," *Digital Investigation*, vol. 7, no. 12, pp. 14 – 27, 2010.
5. S. S. Silva, R. M. Silva, R. C. Pinto, and R. M. Salles, "Botnets: A survey," *Computer Networks*, vol. 57, no. 2, pp. 378 – 403, 2013.
6. Z. Zhu, G. Lu, Y. Chen, Z. Fu, P. Roberts, and K. Han, "Botnet research survey," in *Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International*, July 2008, pp. 967–972.
7. P. Wang, S. Sparks, and C. C. Zou, "An advanced hybrid peer-to-peer botnet," in *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*. USENIX Association, 2007, pp. 2 – 2.
8. D. Dong, Y. Wu, L. He, G. Huang, and G. Wu, "Deep analysis of intending peer-to-peer botnet," in *Grid and Cooperative Computing, 2008. GCC '08. Seventh International Conference on*, Oct 2008, pp. 407–411.
9. I. Ullah, N. Khan, and H. Aboalsamh, "Survey on botnet: Its architecture, detection, prevention and mitigation," in *Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on*, April 2013, pp. 660–665.
10. S. Nagaraja, "Botyacc: Unified p2p botnet detection using behavioural analysis and graph analysis," in *Computer Security - ESORICS 2014*, vol. 8713. Springer International Publishing, 2014, pp. 439–456.
11. S. Garca, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers Security*, vol. 45, pp. 100 – 123, 2014.
12. P. Narang, S. Ray, C. Hota, and V. Venkatakrishnan, "Peershark: Detecting peer-to-peer botnets by tracking conversations," in *2014 IEEE Security and Privacy Workshops*. IEEE Computer Society, 2014, pp. 108–115.
13. P. Narang, V. Khurana, and C. Hota, "Machine-learning approaches for p2p botnet detection using signal-processing techniques," in *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, ser. DEBS '14. ACM, 2014, pp. 338–341.
14. K. Wang, C.-Y. Huang, S.-J. Lin, and Y.-D. Lin, "A fuzzy pattern-based filtering algorithm for botnet detection," *Computer Networks*, vol. 55, no. 15, pp. 3275 – 3286, 2011.
15. J. Ersson and E. Moradian, "Botnet detection with event-driven analysis," *Procedia Computer Science*, vol. 22, no. 0, pp. 662 – 671, 2013.
16. H. Choi and H. Lee, "Identifying botnets by capturing group activities in {DNS} traffic," *Computer Networks*, vol. 56, no. 1, pp. 20 – 33, 2012.
17. H. Choi, H. Lee, H. Lee, and H. Kim, "Botnet detection by monitoring group activities in dns traffic," in *Computer and Information Technology, 2007. CIT 2007. 7th IEEE International Conference on*, Oct 2007, pp. 715–720.
18. H. Choi, H. Lee, and H. Kim, "Botgad: Detecting botnets by capturing group activities in network traffic," in *Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE*. ACM, 2009, pp. 2:1–2:8.
19. S. E. Mathew, A. Ali, and J. Stephen, "Genetic algorithm based layered detection and defense of http botnet," *ACEEE Int. J. on Network Security*, vol. 5, no. 1, pp. 50 – 61, 2014.
20. M. Eslahi, H. Hashim, and N. Tahir, "An efficient false alarm reduction approach in httpbased botnet detection," in *Computers Informatics (ISCI), 2013 IEEE Symposium on*, April 2013, pp. 201–205.
21. C.-Y. Huang, "Effective bot host detection based on network failure models," *Computer Networks*, vol. 57, no. 2, pp. 514 – 525, 2013.

9/25/2016