International Journal of Advances in Computer Networks and Its Security– IJCNS
Volume 5: Issue 2 [ISSN : 2250-3757]
Publication Date : 30 October, 2015

A Survey on Botnet Command and Control Traffic Detection

Ibrahim Ghafir, Jakub Svoboda, Vaclav Prenosil

***Abstract*—Internet users have been attacked by widespread email viruses earlier, but now scenario has been changed. Now attackers are no more interested to just attract media attention by infecting a large number of computers on the network; in fact, their interest has been shifted to compromising and controlling the infected computers for their personal profits. This new attack trend brings the concept of botnets over the global network of computers. With the high reported infection rates, the vast range of illegal activities and powerful comebacks, botnets are one of the main threats against the cyber security. This paper provides the readers with a background on botnet life-cycle, architecture and malicious activities. It also classifies botnet detection techniques, reviews the recent research works on botnet traffic detection and finally indicates some challenges posed to future work on botnet detection.**

***Keywords*—Cyber security, malware, botnet, C&C server, intrusion detection system.**

I. Introduction

The convenience and speed of digital communications have become an integral part of home computer use, as well as every other aspect of use from education to business and research. While high-speed computer networking and the Internet have brought great convenience, a number of security challenges have also emerged with these technologies [1], [2]. Amongst different computer network security threats like viruses and worms, botnets have become one of the most malicious threats over the Internet [3].

A botnet is a collection of computers connected to the Internet which have been compromised and are being controlled remotely by an intruder (the botmaster) via malicious software called bots [4]. Botnets have been used by cyber-criminals to conduct many malicious activities, such as sending spam emails [5], launching DOS attacks [6] and stealing private data [7], [8]. Financial gains are usually the motive for the design and development of botnets by botmasters, who can reportedly make large sums by marketing their technical services [9]. Experts believe that approximately 16-25% of the computers connected to the Internet are members of botnets [10], [11].

Botnets and their detection has been an active area of research in recent times. Many detection techniques have been proposed based on honeynets, network traffic, host-based logs and so on. This paper provides the readers with a background on botnet life-cycle, architecture and malicious activities. It also classifies botnet detection techniques, reviews the recent research works on botnet traffic detection and finally indicates some challenges posed to future work on botnet detection.

The remainder of this paper is organized as follows. Section 2 presents a background on botnets. Botnets detection techniques are classified in Section 3. Section 4 shows some challenges of botnet detection and section 5 concludes the paper.

II. Background

Botnets are networks formed by ”enslaving” host computers, called bots (derived from the word robot), that are controlled by one or more attackers, called botmasters, with the intention of performing malicious activities [12]. In other words, bots are malicious codes running on host computers that allow botmasters to control the host computers remotely and make them perform various actions [13]. The primary purpose of botnets is for the controlling criminal, group of criminals or organized crime syndicate to use hijacked computers for fraudulent online activity.

Like the previous generations of viruses and worms, a bot is a self-propagating application that infects vulnerable hosts through exploit activities to expand their reach. Bot infection methods are similar to other classes of malware that recruit vulnerable systems by exploiting software vulnerabilities, trojan insertion, as well as social engineering techniques leading to download malicious bot code [14]. According to measurement studies in [15] modern bots are equipped with several exploit vectors to improve opportunities for exploitation.

A. *Botnet Life-cycle*

A typical botnet can be created and maintained in five phases including: initial infection, secondary injection, connection, malicious command and control, update and maintenance [16]. This life-cycle is depicted in Figure 1.

During the initial infection phase, the attacker scans a target subnet for known vulnerability, and infects victim machines through different exploitation methods. After initial infection, in secondary injection phase, the infected hosts execute a script known as shell-code. The shell-code fetches the image of the actual bot binary from the specific location via FTP, HTTP, or P2P. The bot binary installs itself on the target machine. Once the bot program is installed, the victim

Ibrahim Ghafir [1], Jakub Svoboda [2], Vaclav Prenosil [1]

[1] Faculty of Informatics, Masaryk University
[2] Institute of Computer Science, Masaryk University
Brno, Czech Republic

computer turns to a ”Zombie” and runs the malicious code. The bot application starts automatically each time the zombie is rebooted [17].

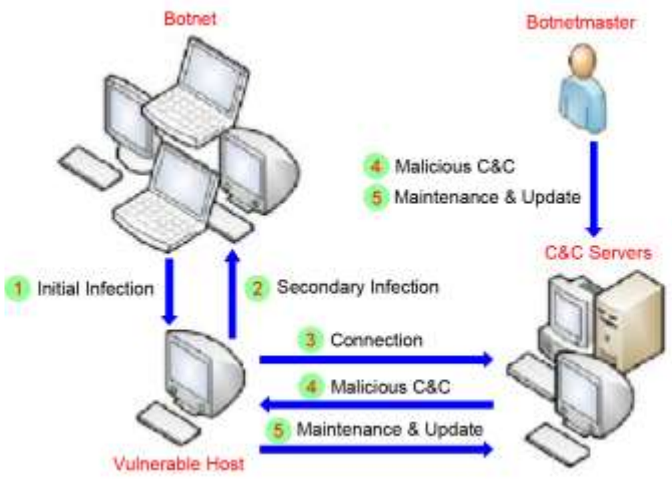

Figure. 1. A typical botnet life-cycle.

In connection phase, the bot program establishes a C&C channel, and connects the zombie to the C&C server. Upon the establishment of C&C channel, the zombie becomes a part of attacker’s botnet army [18]. After connection phase, the actual botnet command and control activities will be started. The botmaster uses the C&C channel to disseminate commands to his bot army. Bot programs receive and execute commands sent by botmaster. The C&C channel enables the botmaster to remotely control the action of large number of bots to conduct various illicit activities [19].

Last phase is to maintain bots live and updated. In this phase, bots are commanded to download an updated binary. Bot controllers may need to update their botnets for several reasons. For instance, they may need to update the bot binary to evade detection techniques, or they may intend to add new functionality to their bot army. Moreover, sometimes the updated binary move the bots to a different C&C server. This process is called server migration and it is very useful for botmasters to keep their botnet alive [20], [21]. Botmasters try to keep their botnets invisible and portable by using Dynamic DNS (DDNS) [22] which is a resolution service that facilitates frequent updates and changes in server locations.

B. *Botnet Architecture*

The command and control channel (C&C), the means by which individual bots form a botnet, may be classified according to its specific architecture [23] and operational modes [24], whether it is centralized, decentralized, hybrid or random architectures, and persistent or periodic modes.

1) Centralized C&C: The centralized approach is similar to the classic client-server network model. Typical examples of this type of botnet architecture are those implemented through the Internet Relay Chat (IRC) protocol [25]. In a centralized C&C infrastructure, all bots establish their communication channel with one, or a few, connection points. These are usually C&C servers that are responsible for sending commands to bots and to provide malware updates.

2) Decentralized C&C: Modern botnets require great flexibility and robustness to be able to handle large numbers of bots and to maximize profits. In [14], decentralized and random architecture for the C&C channel was addressed for the first time. Such decentralized botnets are based on a variety of P2P protocols and work as an overlay network [26].

3) Hybrid model C&C: Hybrid architectures employ characteristics from both centralized and decentralized botnets. In [27], the proposed architecture has the following features; servant bots are the only candidates that can have their IP addresses on the peer lists. They listen to a determined port for incoming connections and use a self-generated symmetric encryption key for communication, which makes botnet detection more difficult. When a bot receives new commands that it has not previously observed, it quickly forwards the command to all servant bots on its peer list.

4) Random model C&C: The random model was introduced by Cooke et al. [14] as a model for future botnets that wish to operate for a long time. In this proposal, the bot does not actively contact the botmaster or other bots. Instead, it waits for connection attempts by the botmaster. To perform an attack, the botmaster scans the network to find zombies, and if it finds one, it sends commands to the bot.

C. *Botnet Malicious Activities*

Botnet can be used for a wide variety of illegitimate activity [28]. They can be exploited for criminally purposes or just for fun, depending on the individuals.

1) Compromising new hosts: To make the botnets stronger, the Botmaster often recruit new hosts using social engineering and distribution of malicious emails.

2) Denial of service attack: DDoS attack capability is a common feature of the botnet. The botnet always contains a set of flooding mechanisms, such as SYN flood, ICMP flood, and HTTP flood, for sending those packets to the targeted network, or just sending thousands of legitimate http, ftp requests to the site.

3) Spam: Spam bots can use an SMTP server to send spam on attacker’s will. Most of today’s e-mail spam is sent by botnet. Phatbot is one such bot widely being used for spamming.

4) Phishing: In most cases, bots can be used for hosting phishing sites. Attackers can extract information from bots by turning them into web servers or DNS servers to conduct phishing.

5) Steal sensitive data: With screen capture, password theft, file upload and key-logging software, botmaster can easily get enough victims’ passwords and information. For example, the SDBot uses advanced key-logging software to collect personal information.

III. Botnet Detection

Botnet C&C traffic is difficult to be detected because: *(1)* it follows normal protocol usage and is similar to normal traffic, *(2)* the traffic volume is low, *(3)* there may be very few bots in the monitored network, and *(4)* may contain encrypted communication [29].

Botnet detection and tracking has been a major research topic in recent years. Different solutions have been proposed in academia. There are mainly two approaches for botnet detection and tracking [17]. One approach is based on setting up honeynets, which is mostly useful to understand botnet technology and characteristics, but does not necessarily detect bot infection. The other approach for botnet detection is based on passive network traffic monitoring and analysis. Botnet detection techniques based on passive traffic monitoring have been useful to identify the existence of botnets. Based on detection method, these techniques can be classified as being signature-based and anomaly-based. Another classification based on audit source location can categorize these techniques into host-based and network-based. All botnet detection techniques will be described and summarized in this section respectively.

A. *Honeypot-based Detection*

Honeypot refers to a decoy system to entice the attention of attackers to attack this computer system to having an aim of protecting the critical targets [30]. Honeypots are computer systems which don't have any production value. According to this concept, a resource that expects no data, so any traffic to or from it is most likely suspicious activity and must be investigated [31], [32]. This technique is very effective for gathering compact high value information such as signature of bots for content-based detection, information of botnet C&C mechanism/servers, unknown security holes that enable bots to penetrate the network [33]. However, honeynets are not necessary able to detect bot infection [34].

In [35], Baecher et al. introduced Nepenthes, a new type of honeypot that inherits the scalability of low-interaction honeypots but at the same time offers a high degree of expressiveness. Nepenthes is a platform to deploy honeypot modules (called vulnerability modules). This is the key to increased expressiveness: Vulnerability modules offer a highly flexible way to configure Nepenthes into a honeypot for many different types of vulnerabilities. In classical terms, Nepenthes still realizes a low-interaction honeypot since it emulates the vulnerable services.

Another honeypot-based intrusion detection system was proposed by Artail et al. [36]. The system adjusts to changes in the organizational network based on the dynamic deployment and configuration of low-interaction honeypots (honeyds). The main idea is for the honeyds to be deployed using available unused IP addresses such that the distribution of operating, systems and services they emulate mimics that of the operating systems and services of the production hosts in the network. In the majority of cases, the traffic that is directed to the honeyds will be seamlessly diverted to high-interaction honeypots where hackers engage with real services.

B. *Passive Network Traffic Monitoring and Analysis*

Botnet detection techniques based on passive traffic monitoring have been useful to identify the existence of botnets [37]. Based on detection method, these techniques can be classified as being signature-based and anomaly-based.

1) Signature-based Detection:

The basic idea is to extract feature information on the packets from the traffic and march the patterns registered in the knowledge base of existing bots [38]. Apparently, it is easy to carry on by simply comparing every byte in the packet, but it also goes with several drawbacks [39]. Firstly, it is unable to identify the undefined bots. Second, it should always update the knowledge base with new signatures, which enhances the management cost and reduces the performance. Third, new bots may launch attacks before the knowledge base are patched.

Snort is a signature-based intrusion detection system [40]; it is basically a combination of multiple components. All the component work together to find a particular attack and then take the corresponding action that is required for that particular attack. A *packet decoder* captures packets from network interfaces and setup the packets to be preprocessed or to be sent to the detection engine. A *preprocessor* captures the raw packet and check them against certain plug-ins. These plug-ins check for a certain type of behavior from the packets. The preprocessor detects anomalies in packet headers and then generate alerts. Once packets have been handled by all enabled preprocessors, they are handed off to the *detection engine* to be checked through a set of rules.

Another signature-based botnet detection software (Rishi) was proposed by Goebel and Holz [41]. This software matches known nick-name patterns of IRC bots. Rishi is primarily based on passive traffic monitoring for suspicious IRC nicknames, IRC servers, and uncommon server ports. It uses n-gram analysis and a scoring system to detect bots that use uncommon communication channels, which are commonly not detected by classical intrusion detection systems. However, Rishi cannot detect encrypted communication as well as non-IRC Botnets. Moreover, this method is unable to detect bots without using known nickname patterns.

In [42], N-EDPS, a signature-based botnet detection and prevention system, was developed by Behal et al. The system focuses on detecting and preventing malware infections (specifically bots/botnets) through monitoring the outbound traffic. The authors utilize the existing open source and freely available software; they used BotHunter [43] as the detection engine and Snort Inline [44] as the prevention engine.

2) Anomaly-based Detection:

Anomaly-based detection techniques attempt to detect botnets based on several network traffic anomalies such as high network latency, high volumes of traffic, traffic on unusual ports, and unusual system behavior that could indicate presence of malicious bots in the network [45]. However, anomaly-based detection techniques have high false alarm rate and the complexity involved in determining what features should be learned in the training phase. What's more, there are no anomalies of botnet until the botnet has been used [46].

Wurzinger et al. [47] use anomaly detection on aggregate network features to identify a deviation from normal activity. Once identified, a snapshot of the network traffic surrounding the anomaly is taken. Using the intuition that snapshots containing similar anomalies are likely multiple instances of a bot responding to the same botmaster command, the packet payloads leading up to the anomaly are searched for common content to find the command. Once a suitable representation of the command is found, the IDS can build a profile which can then be used to detect future occurrences of the command/response pair.

In [29], Gu et al. proposed an approach that uses network-based anomaly detection to identify botnet C&C channels in a local area network without any prior knowledge of signatures or C&C server addresses. This detection approach can identify both the C&C servers and infected hosts in the network. Their approach is based on the observation that, because of the preprogrammed activities related to C&C, bots within the same botnet will likely demonstrate spatial-temporal correlation and similarity.

Binkley and Singh [48] presented an effective algorithm that combines TCP-based anomaly detection with IRC tokenization and IRC message statistics to create a system that can clearly detect client botnets. This algorithm can also reveal bot servers. However, Binkley's approach could be easily defeated by simply using a trivial cipher to encode the IRC commands.

Another algorithm for detection and characterization of botnets was proposed by Karasaridis et al. [49]. It uses passive analysis based on flow data in transport layer. This algorithm can detect encrypted botnet communications. It helps to quantify size of botnets, identify and characterize their activities without joining the botnet.

Network traffic monitoring and analysis techniques used for botnet detection can be also categorized based on audit source location. These categories are host-based and network-based.

1) Host-based Detection:

Host-based systems focus on detecting bot infections on an individual host and typically use signature- or behavior-based methods to correlate network traffic or system events with known bot signatures or behavioral information [50]. While host-based techniques are able to detect single bot infections, some knowledge of the bots behavior must be known in advance. Host-based approaches also benefit from being easy to deploy and from empowering the end-user directly [51].

In 2014, Balram and Wilscy [52] proposed a detection mechanism for bot C&C traffic by analyzing ”suspicious” flows created after filtering out normal traffic from the traffic generated on a host. The filtering is based on a normal profile of the traffic generated by a user on a host. The profile is built dynamically by examining the behavioral pattern of flows to all destinations. A characterization of bot C&C behavior is also proposed, to derive a set of distinguishing attributes based on which detailed analysis is to be done.

In [53], Takemori et al. presented a system which monitors outbound packets from a host and compares with destination-based whitelists. The whitelists are generated by observing an un-infected PC. Although this is a straightforward technique, the detection can be done only during the non-operating time of the PC.

The work in [54], proposed by H Xiong et al., is a host-based bot detection system for HTTP traffic. The detection system is based on the assumption that users have low diversity in the web sites. Out-of-band retrieval and analysis of requested web page is done. Only white-listed web page requests are permitted. The user is informed and asked to take a decision about non white-listed requests. This would be intrusive to the user.

Fedynyshyn et al. [50] presented a host-based method for detecting and differentiating different types of botnet infections based on their C&C styles, e.g., IRC-based, HTTP-based or peer-to-peer (P2P) based. Furthermore, their detection system is completely independent of the content of the C&C messages, i.e., they do not examine packet payloads. The ability to locate and classify botnet C&C connections depends on a few hypotheses; (1) botnet C&C communication can be differentiated from botnet non-C&C communication, (2) botnet C&C communication can be differentiated from legitimate communication, and (3) the characteristics of different styles of C&C are similar across different botnet families.

Another host-based system proposed by Giroire et al. [55] is based on the intuition that bots must contact their C&C server regularly to receive commands from the botmaster. Thus, unlike transient connections, the connections to C&C channels will appear to be persistent. This system first builds a whitelist of legitimate destinations that the monitored host contacts persistently. If any new connection is observed that exhibits high enough temporal persistence, an alarm is raised. If this connection is legitimate, a user can simply add it to the whitelist, otherwise, the connection is assumed to be malicious and is blocked. The success of such a system relies on the assumption that the whitelist is easy to maintain and that it does not need to be updated frequently.

2) Network-based Detection:

Network-based methods attempt to detect bot infections by correlating similar behaviors among several different hosts on the monitored network. Network-based methods do not need prior knowledge of bot signatures or behavioral information as they rely on the intuition that hosts infected by the same bot will behave very similarly to one another whereas uninfected hosts will exhibit different network characteristics from one another [50]. While network-based detection systems may not require prior knowledge of a bot's behavioral patterns, they do require that multiple hosts in the same network become infected for the intrusion to be detectable. In addition, network-based approaches may require additional cooperation of the network administrator and care must be taken to protect the privacy of the network users [56].

BotMiner, proposed by Gu et al. [57], is a network-based botnet detection system. It relies on the group behavior of individual bots within a botnet for its detection. It exploits the underlying uniformity of behavior of botnets and detects them by attempting to observe and cluster similar behavior being performed simultaneously on multiple machines on a network. BotMiner first clusters similar communication activities in the so-called C-plane (for C&C communication traffic). Flows with known safe signatures (such as for some popular protocols) are filtered out of their list to improve performance. Once similar flows have been identified, BotMiner clusters in the so-called A-Plane (for activity traffic) flows by the type of activities they represent using anomaly detection via Snort. By examining both the A-Plane and C-Plane, BotMiner correlates hosts which exhibit both similar network characteristics as well as malicious activity and in doing so identify the presence of a botnet as well as members of the network.

Another network-based detection system, called TAMD, was described by Yen and Reiter [58]. TAMD is an abbreviation for "Traffic Aggregation for Malware Detection". As its name suggests, TAMD distills traffic aggregates from the traffic passing the edge of a network, where each aggregate is defined by certain characteristics that the traffic grouped within it shares in common. By refining these aggregates to include only traffic that shares multiple relevant characteristics, and by using past traffic as precedent to justify discarding certain aggregates as normal, TAMD constructs a small set of new aggregates (i.e., without previous precedent) that it recommends for examination, for example, by more targeted (e.g., signature-based) intrusion detection tools.

IV. Botnet Detection Challenges

There are several challenges facing botnet detection research (and, more generally, research on intrusion detection) [59], [60]. We can summarize theses challenges by the following points:

- *Multiple administrative domains:* The Internet is controlled by many different organizations, which have different goals, interests and policies, and which tend to be guarded about data sharing.

- *Heterogeneity:* Different networks can have widely different characteristics; for example, academic and corporate networks differ considerably [61]. It is difficult to capture the diversity of the Internet with a small number of network traces.

- *Lack of ground truth:* Given a host within a network trace, it is difficult to establish whether or not it is part of a botnet. This is particularly true for hosts in other administrative domains where researchers cannot directly investigate.

- *Privacy concerns:* Network traces contain sensitive information about the actions and communications of the users of the network; thus, it is difficult to persuade network operators to collect them, let alone share them with a third party.

V. Conclusion

A great deal of recent research has examined botnets; despite some real advances, few results have been adopted and implemented in real network scenarios. This survey provides the readers with a background on botnet life-cycle, architecture and malicious activities. It also classifies botnet detection techniques, reviews the recent research works on botnet traffic detection and finally indicates some challenges posed to future work on botnet detection.

No technique is perfect and each detection approach comes with its own unique set of trade-offs with respect to false positives and false negatives. Therefore, detecting and tracking compromised hosts in a botnet will continue to be a challenging task. For future work, we are working on developing a network-based botnet detection system based on the correlation between the outputs of different detection methods. Each detection method aims to detect one possible technique used in botnet C&C communication.

References

[1] J.-S. Lee, H. Jeong, J.-H. Park, M. Kim, and B.-N. Noh, "The activity analysis of malicious http-based botnets using degree of periodic repeatability," in Security Technology, 2008.

[2] M. La Polla, F. Martinelli, and D. Sgandurra, "A survey on security for mobile devices," Communications Surveys & Tutorials, IEEE, vol. 15, no. 1, pp. 446–471, 2013.

[3] L. Zhang, S. Yu, D. Wu, and P. Watters, "A survey on latest botnet attack and defense," in Trust, Security and Privacy in Computing and Communications (TrustCom), 2011.

[4] B. Al-Duwairi and L. Al-Ebbini, "Botdigger: a fuzzy inference system for botnet detection," in Internet Monitoring and Protection (ICIMP), 2010 Fifth International Conference on. IEEE, 2010, pp. 16–21.

[5] A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers," ACM SIGCOMM Computer Communication Review, vol. 36, no. 4, pp. 291–302, 2006.

[6] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring internet denial-of-service activity," ACM Transactions on Computer Systems (TOCS), vol. 24, no. 2, pp. 115–139, 2006.

[7] S. Saroiu, S. D. Gribble, and H. M. Levy, "Measurement and analysis of spyware in a university environment." in NSDI, 2004, pp. 141–153.

[8] T. Holz, M. Engelberth, and F. Freiling, Learning more about the underground economy: A case-study of keyloggers and dropzones. Springer, 2009.

[9] R. A. Rodr´ıguez-G´omez, G. Maci´a-Fern´andez, and P. Garc´ıa Teodoro, "Survey and taxonomy of botnet research through life-cycle," ACM Computing Surveys (CSUR), vol. 45, no. 4, p. 45, 2013.

[10] W. Sturgeon, "Net pioneer predicts overwhelming botnet surge," ZDNet News, January, vol. 29, 2007.

[11] B. AsSadhan, J. M. Moura, D. Lapsley, C. Jones, and W. T. Strayer, "Detecting botnets using command and control traffic," in Network Computing and Applications, 2009.

[12] P. Bacher, T. Holz, M. Kotter, and G. Wicherski, "Know your enemy: Tracking botnets," 2005.

[13] 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, ACM, 2009.

[14] E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: Understanding, detecting, and disrupting botnets," in Proceedings of the USENIX SRUTI Workshop, vol. 39, 2005, p. 44.

[15] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in Proceedings of

the 6th ACM SIGCOMM conference on Internet measurement. ACM, 2006, pp. 41–52.

[16] M. Feily, A. Shahrestani, and S. Ramadass, “A survey of botnet and botnet detection,” in Emerging Security Information, Systems and Technologies, 2009. SECURWARE’. 2009, pp. 268–273.

[17] 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. IEEE, 2008, pp. 967–972.

[18] R. Villamar´ın-Salom´on and J. C. Brustoloni, “Identifying botnets using anomaly detection techniques applied to dns traffic,” in Consumer Communications and Networking Conference, 2008, pp. 476–481.

[19] K.-K. R. Choo, Zombies and botnets. Australian Institute of Criminology, 2007.

[20] D. Dagon, G. Gu, C. P. Lee, and W. Lee, “A taxonomy of botnet structures,” in Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual. IEEE, 2007, pp. 325–339.

[21] 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, pp. 715–720.

[22] J. Bound and Y. Rekhter, “Dynamic updates in the domain name system (dns update),” 1997.

[23] T. Micro, “Taxonomy of botnet threats,” Whitepaper, November, 2006.

[24] L. Liu, S. Chen, G. Yan, and Z. Zhang, “Bottracer: Execution-based bot-like malware detection,” in Information Security. Springer, 2008, pp. 97–113.

[25] C. Kalt, “Internet relay chat: Architecture,” http://tools.ietf.org/html/ rfc2810, accessed: 2015-2-20.

[26] M. Jelasity and V. Bilicki, “Towards automated detection of peer-topeer botnets: On the limits of local approaches,” in USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), 2009.

[27] P. Wang, S. Sparks, and C. C. Zou, “An advanced hybrid peer-topeer botnet,” Dependable and Secure Computing, IEEE Transactions on, vol. 7, no. 2, pp. 113–127, 2010.

[28] C. Li, W. Jiang, and X. Zou, “Botnet: Survey and case study,” in Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on. IEEE, 2009, pp. 1184–1187.

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

[30] S. Kumar, R. Sehgal, P. Singh, and A. Chaudhary, “Nepenthes honeypots based botnet detection,” arXiv preprint arXiv:1303.3071, 2013.

[31] J. Baltazar, J. Costoya, and R. Flores, “Infiltrating waledac botnet’s covert operations,” TREND MICRO, 2009.

[32] P. Porras, H. Sa¨ıdi, and V. Yegneswaran, “A foray into confickers logic and rendezvous points,” in USENIX Workshop on Large-Scale Exploits and Emergent Threats, 2009.

[33] H. R. Zeidanloo, M. J. Z. Shooshtari, P. V. Amoli, M. Safari, and M. Zamani, “A taxonomy of botnet detection techniques,” in Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, vol. 2. IEEE, 2010, pp. 158–162.

[34] J. Zhuge, T. Holz, X. Han, J. Guo, and W. Zou, Characterizing the IRC-based botnet phenomenon. Universit¨at Mannheim/Institut f¨ur Informatik, 2007.

[35] P. Baecher, M. Koetter, T. Holz, M. Dornseif, and F. Freiling, “The nepenthes platform: An efficient approach to collect malware,” in Recent Advances in Intrusion Detection. Springer, 2006, pp. 165–184.

[36] H. Artail, H. Safa, M. Sraj, I. Kuwatly, and Z. Al-Masri, “A hybrid honeypot framework for improving intrusion detection systems in protecting organizational networks,” computers & security, vol. 25, no. 4, pp. 274–288, 2006.

[37] S. Garc´ıa, A. Zunino, and M. Campo, “Survey on network-based botnet detection methods,” Security and Communication Networks, vol. 7, no. 5, pp. 878–903, 2014.

[38] J. Liu, Y. Xiao, K. Ghaboosi, H. Deng, and J. Zhang, “Botnet: classification, attacks, detection, tracing, and preventive measures,” in EURASIP journal on wireless communications and networking, vol. 2009. IEEE Computer Society, 2009, pp. 1184–1187.

[39] Y. Kugisaki, Y. Kasahara, Y. Hori, and K. Sakurai, “Bot detection based on traffic analysis,” in Intelligent Pervasive Computing, 2007. IPC. The 2007 International Conference on. IEEE, 2007, pp. 303–306.

[40] P. Agarwal and S. Satapathy, “Implementation of signature-based detection system using snort in windows,” 2014.

[41] G. J. Rishi, “identify bot contaminated hosts by irc nickname evaluation,” Cambridge, MA: Proceedings of the HotBots, vol. 7.

[42] S. Behal, A. S. Brar, and K. Kumar, “Signature-based botnet detection and prevention,” http://www. rimtengg. com/iscet/proceedings/pdfs/advcom p/148. pdf, 2010.

[43] G. Gu, P. A. Porras, V. Yegneswaran, M. W. Fong, and W. Lee, “Bothunter: Detecting malware infection through ids-driven dialog correlation.” in Usenix Security, vol. 7, 2007, pp. 1–16.

[44] P. S. Kenkre, A. Pai, and L. Colaco, “Real time intrusion detection and prevention system,” in Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing, Springer, 2015, pp. 405–411.

[45] B. Saha and A. Gairola, “Botnet: an overview,” CERT-In White Paper, CIWP-2005-05, vol. 240, 2005.

[46] I. Ullah, N. Khan, and H. A. Aboalsamh, “Survey on botnet: Its architecture, detection, prevention and mitigation,” in Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on. IEEE, 2013, pp. 660–665.

[47] P. Wurzinger, L. Bilge, T. Holz, J. Goebel, C. Kruegel, and E. Kirda, “Automatically generating models for botnet detection,” in Computer Security–ESORICS 2009. Springer, 2009, pp. 232–249.

[48] J. R. Binkley and S. Singh, “An algorithm for anomaly-based botnet detection,” in Proceedings of USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI), 2006, pp. 43–48.

[49] A. Karasaridis, B. Rexroad, and D. Hoeflin, “Wide-scale botnet detection and characterization,” in Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, vol. 7. Cambridge, MA, 2007.

[50] G. Fedynyshyn, M. C. Chuah, and G. Tan, “Detection and classification of different botnet c&c channels,” in Autonomic and Trusted Computing. Springer, 2011, pp. 228–242.

[51] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, “A survey of intrusion detection techniques in cloud,” Journal of Network and Computer Applications, vol. 36, no. 1, pp. 42–57, 2013.

[52] S. Balram and M. Wilscy, “User traffic profile for traffic reduction and effective bot c&c detection.” IJ Network Security, vol. 16, no. 1, pp. 46–52, 2014.

[53] K. Takemori, M. Nishigaki, T. Takami, and Y. Miyake, “Detection of bot infected pcs using destination-based ip and domain whitelists during a non-operating term,” IEEE GLOBECOM 2008, pp. 1–6.

[54] H. Xiong, P. Malhotra, D. Stefan, C. Wu, and D. Yao, “User-assisted host-based detection of outbound malware traffic,” in Information and Communications Security. Springer, 2009, pp. 293–307.

[55] F. Giroire, J. Chandrashekar, N. Taft, E. Schooler, and D. Papagiannaki, “Exploiting temporal persistence to detect covert botnet channels,” in Recent Advances in Intrusion Detection. Springer, 2009, pp. 326–345.

[56] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, “Intrusion detection system: A comprehensive review,” Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16–24, 2013.

[57] G. Gu, R. Perdisci, J. Zhang, W. Lee et al., “Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection.” in USENIX Security Symposium, vol. 5, no. 2, 2008, pp. 139–154.

[58] T.-F. Yen and M. K. Reiter, “Traffic aggregation for malware detection,” in Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, 2008, pp. 207–227.

[59] A. J. Aviv and A. Haeberlen, “Challenges in experimenting with botnet detection systems.” in CSET, 2011.

[60] 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.

[61] H. Pucha, Y. C. Hu, and Z. M. Mao, “On the impact of research network based testbeds on wide-area experiments” in Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, 2006, pp. 133–146.