THE JOURNAL OF DIGITAL FORENSICS, SECURITY AND LAW

Journal of Digital Forensics, Security and Law

Volume 10 | Number 1 Article 2

2015

A Survey of Botnet Detection Techniques by Command and Control Infrastructure

Thomas S. Hyslip
Norwich University

Jason M. Pittman
Cal Poly Pomona University

A SURVEY OF BOTNET DETECTION TECHNIQUES BY COMMAND AND CONTROL INFRASTRUCTURE

Thomas S. Hyslip, Sc.D.
Norwich University
919-274-4526
thyslip@norwich.edu

Jason M. Pittman, Sc.D.
Cal Poly Pomona University
805-907-5313
jmpittman@cpp.edu

ABSTRACT

Botnets have evolved to become one of the most serious threats to the Internet and there is substantial research on both botnets and botnet detection techniques. This survey reviewed the history of botnets and botnet detection techniques. The survey showed traditional botnet detection techniques rely on passive techniques, primarily honeypots, and that honeypots are not effective at detecting peer-to-peer and other decentralized botnets. Furthermore, the detection techniques aimed at decentralized and peer-to-peer botnets focus on detecting communications between the infected bots. Recent research has shown hierarchical clustering of flow data and machine learning are effective techniques for detecting botnet peer-to-peer traffic.

Keywords: botnet, botnet detection, distributed denial of service, malware

1. INTRODUCTION

The term 'botnet' is now associated with cybercrime and hacking (Alhomoud, Awan, Disso, & Younas, 2013). However, botnets were originally developed to assist with the administration of Internet Relay Chat (IRC) Servers (Cooke et al., 2005). As the popularity of IRC expanded, the IRC server administrators developed software to perform automated functions to assist with the administration of the IRC Servers (Cooke et al., 2005). The computers that operated the software and performed the automated functions were referred to as robot computers and eventually as bots (Dittrich, 2012). The Eggdrop IRC bot was the first IRC Bot, developed in 1993 by Jeff Fisher to assist with the administration of IRC channels and which is still in use today (Alhomoud et al., 2013; Cooke et al., 2005). Eventually, a network of bots was developed under the direction of IRC administrators and became known as a botnet (Dittrich, 2012). IRC administrators were able to send a single command from their computer and the botnet would execute that command on all the IRC Servers. Figure 1 shows a typical network configuration of an IRC botnet. Nefarious individuals realized the potential of botnets for unethical purposes and the botnets began to infect IRC users' computers without the users' knowledge and use those computers without the users' consent (Cao & Qiu, 2013; Cooke et al, 2005).

A Computer Emergency Response Team, Coordination Center (CERT/CC) advisory published on March 11, 2003, CERT/CC Advisory CA-2003-08, warned against the GT-bot and sdbot utilizing IRC to remotely control compromised systems (Householder & Danyliw, 2003). Householder and Danyliw

(2003) also highlighted the growing size of botnets, with reports of GT-bot botnets in excess of 140,000 bots and the sdbot with over 7000 compromised systems. Householder and Danyliw also warned of the botnets' ability to launch distributed denial of service attacks with TDP, UDP, and ICMP packets.

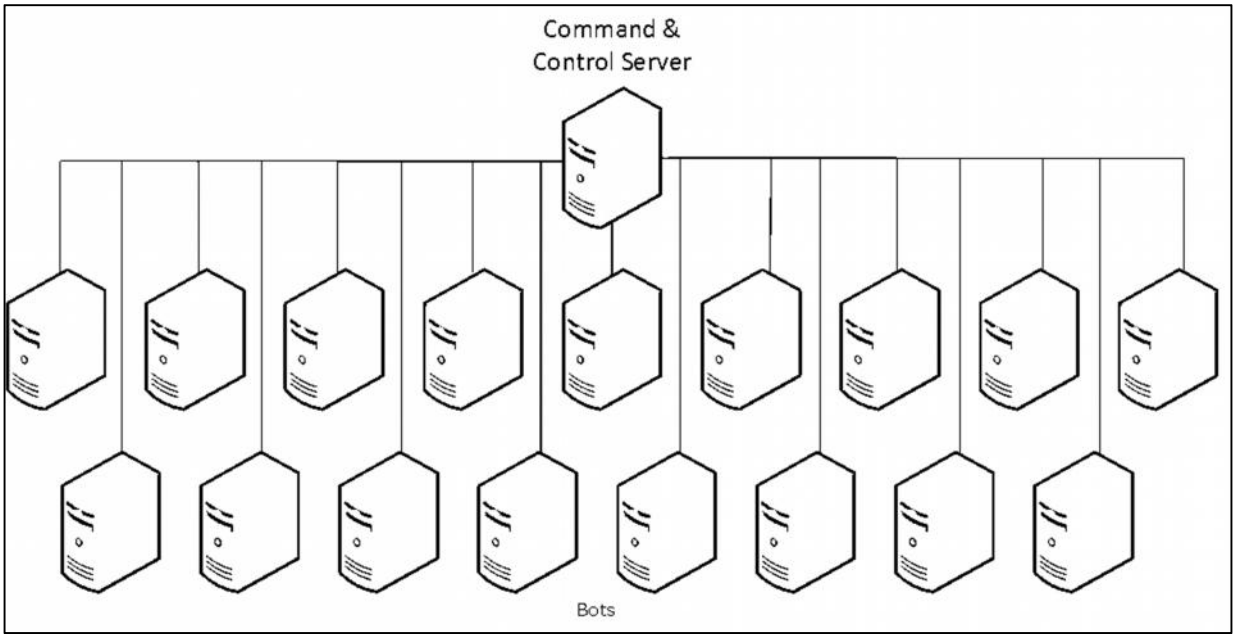

Figure 1. An IRC Botnet diagram showing the individual connections between each "bot" and the command and control server.

The size and scope of botnets continued to rise at an alarming rate and in February 2010, Spanish authorities and the FBI dismantled the Mariposa botnet, which consisted of over 12 million compromised computers (Roscini, 2014). Only 2 years after the takedown of the Mariposa botnet, another botnet, the Metulji botnet, was dismantled by the FBI and consisted of over 20 million compromised computers (Ventre, 2013). In 2013, Rossow and Dietrich considered botnets to be one of the Internet's most serious threats and Awan et al. (2013) believed botnets are a priority for many countries' cyber defenses.

There has been considerable research into botnets and botnet detection techniques, but botnets are constantly evolving to stay ahead of the latest detection techniques (Brezo, Santos, Bringas, & Val, 2011; Feily, Shahrestani, & Ramadass, 2009; Hasan, Awadi, & Belaton, 2013; Zeng, 2012; Zhang, 2012). This survey analyzed the history and evolution of botnet detection as botnets changed from a centralized command and control structure to a decentralized peer-to-peer control structure. When early research on botnet detection focused on the use of passive honeypots and detection techniques aimed at detecting botnet command and control communications in centralized botnets, Botmasters began to use peer-to-peer and decentralized communications (Feily et al., 2009; Hasan, Awadi, & Belaton, 2013; Zeng, 2012; Zhang, 2012). Botnet detection techniques were then developed to identify communications between infected computers within the decentralized botnets and

Botmasters responded with the use of obfuscated and encrypted communications (Brezo, Santos, Bringas, & Val, 2011; Feily et al., 2009; Gu, Porras, Yegneswaran, Fong, & Lee, 2007; Zeng, 2012; Zhang, 2012).

There have been several previous surveys of botnet detection techniques, but most are dated prior to 2009 and do not include botnet detection techniques aimed at decentralized or encrypted botnets (Feily et al., 2009; Bailey, Cooke, Jahanian, Yunjing, & Karir, 2009; Zhu, Lu, Chen, Fu, Roberts, & Han, 2008). Silva, Silva, Pinto and Salles (2013) conducted a survey of Botnets that included peer to peer, decentralized, and encrypted botnets. Silva et al. included a history of botnets and a survey of different botnet detection techniques, as well as a sample of techniques for botnet defense.

What separates this survey from previous work is the comparison of botnet detection techniques by command and control infrastructure. To the best of our knowledge, previous research has not yet clearly identified which detection techniques are effective against which types of command and control infrastructure. This survey provides a comprehensive review of botnet detection techniques and provides tables for quick review of which techniques are effective against which command and control infrastructures.

2. EARLY BOTNET DETECTION (2005-2010)

The Honeynet project was a pioneer in botnet detection (Feily et al., 2009). The Honeynet project began in 1999 as an information mailing list for information security professionals and was established as a non-profit information security research organization with the mission to learn about computer and network attacks in 2000 (Spitzner, 2003). Spitzner (2003) defined a honeynet as a network of computers placed on the Internet with the intention of capturing unauthorized activity directed at the computers. The purpose of a honeynet is to monitor network activity after malicious software is installed on the honeynet's computers and learn how the malicious software operates, with the goal of capturing new and unknown attacks and malicious software (Spitzner, 2003). In a 2009 survey of botnet detection techniques, Feily et al. (2009) found a vast majority of the botnet detection techniques rely heavily on honeynets because honeynets are simple to operate and are passive to the botnet, so no interaction is required with the botmaster or command and control server by the researcher. The honeynet receives the instructions or commands from the botnet operator but does not itself respond or execute the commands (Spitzner, 2003).

In July 2005, Cooke, Jahanian, and McPherson proposed monitoring transmission control protocol (TCP) port 6667 on live networks for IRC botnet command and control traffic as a possible botnet detection technique. TCP port 6667 is the default IRC port, but Cooke et al. recognized the default port is easily changed to non-standard ports, so the detection technique of monitoring networks for IRC traffic on TCP 6667 was not recommended. Cooke et al. proposed a second botnet detection technique utilizing a honeypot and capturing traffic between the honeypot and the IRC botnet command and control server. The captured traffic was then analyzed to develop signatures of botnet traffic (Cooke et al, 2005). Cooke et al. determined there were no connection-based variables that would be useful in detecting botnets via monitoring network traffic for command and control traffic. The botnets' ability to modify the mode or behavior of communications can easily defeat detection techniques based on command

and control traffic analysis (Cooke et al., 2005).

Although Cooke et al. (2005) determined monitoring for command and control traffic was not effective, Gu et al. (2007) develop BotHunter, to detect inbound command and control traffic with bots inside a local area network. Gu et al. developed two plugins and one ruleset for the open source, intrusion detection system, Snort (Cisco, 2014). For inbound traffic detection, Gu et al. (2007) developed the Snort plugin, Statistical Scan Anomaly Detection (SCADE) which monitors 24 TCP and 4 UDP inbound ports for possible command and control traffic associated with botnet malware. SCADE also monitors outbound traffic for hosts that scan a large number of external IP addresses or have high number of failed external connections.

The second Snort (Cisco, 2014) plugin developed by Gu et al. (2007) Statistical Payload Anomaly Detection Engine (SLADE) attempts to detect malicious payloads through packet inspection of all inbound traffic. SLADE utilizes anomaly detection to determine if payloads are suspicious based on the payloads standard deviation from test payloads of normal Internet traffic (Gu et al., 2007). The problem with deep packet inspection is the large overhead associated with inspecting voluminous amounts of traffic in large networks (Zhang, 2012). Gu et al. (2007) also developed four rulesets for Snort (Cisco, 2014) to monitor 1383 heuristics of known botnets and malware. BotHunter's final phase of detection is a correlation matrix that weighs each Snort alert and applies a coefficient based on the type of alert to determine if a host is infected (Gu et al., 2007).

Gu, Zhang, and Lee (2008) built upon BotHunter to develop BotSniffer, a system designed to detect botnet command and control traffic through anomaly detection. BotSniffer is limited to detecting IRC and HTTP botnets that use a centralized command and control server, but no prior knowledge of a botnet's signature is required to detect hosts within a local area network (Gu, Zhang, et al., 2008). In both IRC and HTTP botnets, Gu, Zhang, et al. recognized that the bots must make connections to the command and control server to obtain commands and then the bots will have similar activity based on the commands. Based on research conducted by Zhuge, Holz, Han, Guo, & Zou (2007), Gu and his associates developed BotSniffer to recognize similar behavior by hosts after communicating with a possible command and control server located at the same IP address. Zhuge et al. (2007) had determined that over 28% IRC botnet commands are for spreading malware and 25% of IRC commands are for distributed denial of service attacks. Based on these statistics, Gu, Zhang et al. (2008) developed anomaly based algorithms to detect command and control traffic, as well as network scanning, with the open source intrusion detection system, Snort (Cisco, 2014). Utilizing previously captured network traffic with known botnet infections, Gu, Zhang et al. (2008) successfully tested BotSniffer and detected 100% of IRC botnet command and control traffic with a false positive rate of 0.16%.

Research by Karasaridis, Rexford, and Hoeflin (2007) in anomaly-based detection techniques demonstrated the ability to calculate the size of botnets as well as identify command and control servers by analyzing flow data from the transport layer in large-scale networks. However, this technique was only tested against IRC based botnets utilizing a centralized command and control server (Karasaridis et al., 2007). Karasaridis et al. recommended additional research in the detection of peer-to-peer and HTTP based botnets.

With the introduction of botnets communicating via peer to peer networks, Gu, Perdisci et al. (2008) developed BotMiner as a botnet detection technique that is effective against any botnet command and control protocol or structure, including peer to peer. Figure 2 shows a typical peer to peer botnet infrastructure without a central command and control server. BotMiner detects botnets by clustering hosts based on similar traffic and malicious activities (Gu, Perdisci et al., 2008). Gu, Perdisci et al.'s research focused on the botnet communications since botnets much communicate with a command and control server of with other bots to receive commands such as when to scan or launch attacks. In order for the bots to function as a botnet, the bots must receive the same commands; therefore the researchers believed the same botnet would have similar traffic and malicious activities (Gu, Perdisci et al., 2008). Based on the similar traffic and activities, BotMiner clusters similar communication traffic into C-plane traffic and like malicious activities into A-plane traffic (Gu, Perdisci et al., 2008). Gu, Perdisci et al. then detected botnets by correlating the A-plane and C-plane traffic.

To cluster communications within the C-plane traffic, Gu, Perdisci, et al. (2008) monitored TCP and UDP network flow data and recorded IP addresses, network ports time and duration of the traffic, and the number of packets and bytes transferred in each direction. Gu, Perdisci et al. used Snort (Cisco, 2014) to capture A-plane traffic based on malicious activities, scanning, spam, and binary downloads. The C-plane clusters were then correlated with the A-plane clusters to identify hosts that are part of a botnet (Gu, Perdisci et al., 2008).

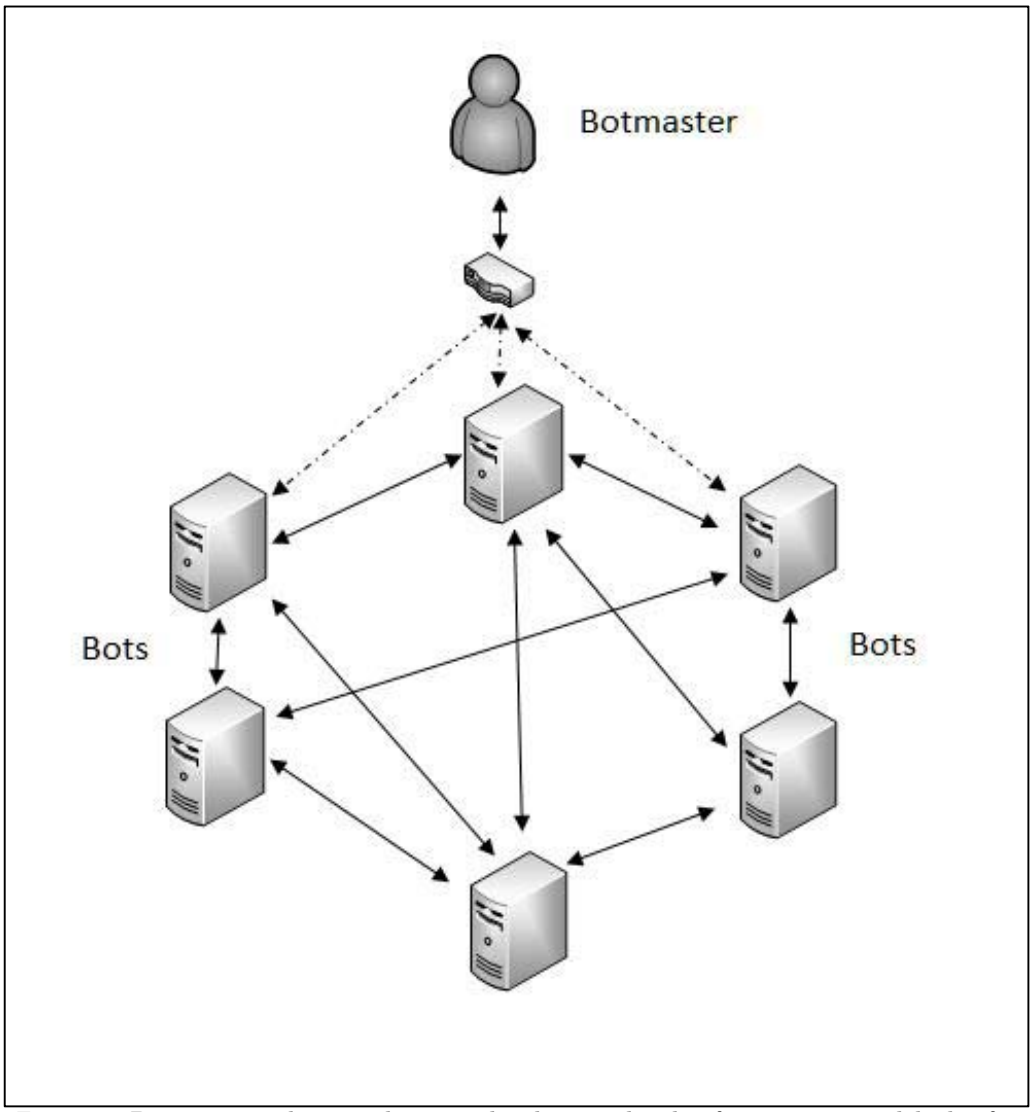

Figure 2. Peer to peer botnet showing the decentralized infrastructure and lack of a command and control server. The Botmaster is able to communicate directly with a bot and the commands are passed between the bots.

Wang and Yu (2009) developed a botnet detection technique aimed at detecting command and control communications of centralized botnets, irrespective of the particular botnet. Wang and Yu based their detection technique on the timing and uniformity of botnet communications; Wang and Yu's technique used only the packet size and timing interval between arriving packets as variables to determine if network traffic was botnet command and control communications. Experimental results showed the technique to be effective for detecting command and control traffic of four different botnet types. However, the technique is only effective against botnets with a centralized command and control structure (Wang & Yu, 2009).

Using structured overlay networks for communication, Nagaraja, Mittal, Hong, Caesar and Borisov (2010) developed BotGrep, a botnet detection technique focused on peer-to-peer botnets. Nagaraja et al. developed an

algorithm that isolates peer-to-peer communication based on the pairing of nodes that communicate with each other. BotGrep then utilizes graph analysis to identify botnet hosts. Although BotGrep is not affected by botnets that vary ports or use encryption, BotGrep does require a seeding of botnet information to be effective; therefore, the researchers recommend operating a honeynet to capture botnet intelligence that can be used by BotGrep to identify the rest of the botnet (Nagaraja et al., 2010).

Prior detection techniques relied on either host level detection or network level detection. Hoever, Zeng, Hu and Shin (2010) developed a botnet detection technique that incorporates both host level detection and network level detection. Zeng et al. believed that by combining the host and network level detections and correlating the alerts, their technique would increase the rate of detection and overcome the limitation of each technique alone. Zeng et al. used registry changes, file system modifications and network stack changes to alert for possible botnet malware activity on host detections and utilized netflow data for network level detection but avoided full packet inspection, which ensures privacy for network users. The researchers successfully tested the combined host and network detection technique. Such may very well be the first combined host and network level detection technique developed. Further, Zeng et al. stated that their combined host detection technique was effective against IRC, peer-to-peer, and HTTP botnets, but noted that the technique is limited by the scalability. Zeng et al. recognized that the host level detection technique requires installation on all hosts within an organization and may only be accomplished in enterprise networks.

Table 1 summarizes early botnet detection techniques based on the techniques ability to detect different types of botnet infrastructure. Table 1 also provides an indirect timeline of botnet infrastructures and communications. While early botnets used IRC exclusively, the introduction of HTTP and P2P communications is evident.

Table 1
Early Botnet Detection Techniques

Researchers	IRC	HTTP	P2P
Cooke et al. (2005)	X		
Gu et al. (2007)	X		
Karasaridis, Rexford, and Hoeflin (2007)	X		
Gu, Zhang, and Lee (2008)	X	X	
Gu, Perdisci et al. (2008)	X	X	X
Wang and Yu (2009)	X	X	
Nagaraja, Mittal, Hong, Caesar and Borisov (2010)			X
Zeng, Hu and Shin (2010)	X	X	X

3. MODERN BOTNET RESEARCH (2011-14)

With the increase in peer-to-peer and decentralized botnets a majority of modern research has focused on detecting peer-to-peer and decentralized botnets, in particular, the communications between bots within the botnet. Francois, Wang, State and Engel (2011) developed BotTrack and overcome the limitations of forensic analysis when examining large datasets of NetFlow data to detect peer-to-peer botnet communications. Similar to BotGrep (Nagaraja et al., 2010), Francois et al. developed BotTrack to identify peer-to-peer connections between hosts and identify botnet hosts utilizing an algorithm and graph analysis. Building on BotTrack, Francois, Wang, Bronzi, State and Engel (2011) used Hadoop (Hadoop, 2013), an open source form of distributed computing based on Google's MapReduce (Dean & Ghemawat, 2004) to develop BotCloud to efficiently analyze NetFlow data. BotCloud showed improved detection rates when prior information about botnets is developed with a honeypot (Francois et al., 2011). Furthermore, BotCloud's use of Hadoop (2013) increased the efficiency and speed of botnet detection (Francois et al., 2011).

Zhang, Perdisci, Lee, Sarfraz and Luo (2011) developed a botnet detection technique to detect botnet peer-to-peer communications utilizing statistical fingerprints of peer-to-peer traffic. Peer-to-peer botnets have an advantage over IRC or HTTP protocol botnets because the former do not have a centralized command and control server and single point of failure (Zhang et al., 2011). The lack of a centralized command and control server make peer-to-peer botnets more resilient and more difficult to disable (Zhang et al., 2011). Zhang et al.'s peer-to-peer detection technique was focused on local area networks (LANS) and enterprise wide area networks (WANS); to detect peer-to-peer botnets. Zhang et al.'s technique first detects all peer-to-peer traffic and hosts and then develops signatures for different applications. Based on the signatures, Zhang et al. were able to differentiate legitimate peer-to-peer traffic from botnet peer-to-peer traffic. To develop the signatures of peer-to-peer traffic, Zhang et al. used the length of time a peer-to-peer program is operating because botnets run as long as possible and whenever a computer is turned on, while legitimate peer-to-peer programs are often started and stopped by the user. Based on the length of time a peer-to-peer program is active, Zhang et al. filtered out peer-to-peer hosts with short active times.

After filtering the peer-to-peer traffic based on length of active peer-to-peer traffic Zhang et al. (2011) further differentiated the traffic based on IP addresses contacted by peer-to-peer hosts. Since peer-to-peer botnet hosts within the same LAN/WAN will often communicate with the same IP addresses and with other bots within the LAN/WAN, the researchers were able to filter out peer-to-peer hosts that did not communicate with any IP addresses that were not contacted by other peer-to-peer hosts (Zhang et al., 2011). The final filter Zhang et al. applied was based on the connection status of the traffic. If a peer-to-peer host had completed an outgoing three way handshake on a TCP connection or a UDP connection with a request and response packet, the traffic is kept and all other traffic is filtered out (Zhang et al., 2011). Zhang et al. based this filter on their findings that peer-to-peer nodes function as both a server and a client, and must accept connections from other hosts in the network and initiate connections with the same hosts. After this traffic filtering was complete, Zhang et al. attempted to identify peer-to-peer botnet hosts.

Zhang et al.'s final action to identify peer-to-peer botnet hosts involved differentiating between legitimate peer-to-peer traffic and

botnet peer-to-peer traffic. To determine this, Zhang et al. analyzed the traffic for hosts that ran the same protocol and communicated with a high percentage of the same IP addresses. As stated earlier, bots of the same peer-to-peer botnet will communicate with each other and share IP destinations of other bots within the botnet. Furthermore, Zhang et al.'s research showed bots of the same botnet use the same peer-to-peer protocol. Based on these filters and detection techniques, Zhang et al. were able to detect 100% of the peer-to-peer bots within captured network traffic with only a 0.2% false positive rate.

As botnets began to use encrypted communications, Barthakur, Dahal and Ghose (2012) developed a procedure for detecting encrypted peer-to-peer botnet communications. Barthakur et al. used Support Vector Machines to analysis network traffic and classify botnet communications based on patterns and statistical differences between peer-to-peer botnet communications and normal web traffic. Barthakur et al. recognized botnet communications use many random ports and attempt to keep packet sizes to a minimum, which is the opposite of legitimate peer-to-peer to traffic. Based on these facts, Support Vector Machines were able to analyze patterns of peer-to-peer traffic and successfully identify botnet communications (Barhakur et al., 2012).

Han, Chen, Xu and Liang (2012) proposed a botnet detection and suppression system called Garlic. Han et al. believed Botmasters attempted to keep botnets as small possible to avoid detection and allow the Botmaster to easily change the botnet's command and control server. Han et al. stated the botnet suppression system, Garlic, was capable of automatically detecting and suppressing botnets. Han et al.'s Garlic suppression system relied on terminal nodes distributed throughout a network and the nodes collaborated with each other to detect patterns and alerts based on rules. Han et al. also observed that Garlic would regenerate rules based on feedback from the alerts and redistributed updated rules to the terminal nodes. During experimental testing, Han et al. were able to detect all 20 bots within 45 minutes; however, they only experimented with IRC botnets operating on TCP ports 6660-6669 (including IRC port 6667), as well as HTTP botnets operating on port 80. Han et al. did not test peer-to-peer botnet nor did they provide any research on peer-to-peer botnets within their study.

Increasingly, botnets expand through drive by download attacks. In response, Zhang (2012) developed a new botnet detection technique to identify drive by download attacks and detect botnets in the infection stage. Zhang recognized that many botnets use drive by downloads to infect new bots and by preventing the initial infection the size and scope of botnets could be greatly diminished. To identify drive by download techniques, Zhang collected HTTP traces from honeypots and whenever exploits were detected, the honeypots used a dynamic WebCrawler to record the URLs and IP addresses of the domains. Zhang then clustered groups of hostnames that share IP addresses. By clustering the hostnames based on shared IP addresses, Zhang was able to defeat the botnets that use fast flux network changes to command control server domain names and IP addresses. Fast flux networks use numerous IP addresses for one domain name and repeatedly update the DNS records for the domain name to different IP addresses to avoid detection (Caglayan, Toothaker, Drapaeau, & Burke, 2010).

Furthermore, Zhang (2012) also developed a system to increase the scalability of botnet detection systems. Zhang's system improved upon current detection systems by reducing

the amount of packets requiring deep packet inspection; Zhang accomplished this by developing a three-step process that captures network flows, correlated the network flows and detected botnets through fine grain analysis. Rather than use deep packet inspection, Zhang's system used network flow information and packet header information, which allowed for deployments in larger networks and the ability to inspect traffic for botnet command and control traffic.

Zhang (2012) also developed a flow-capture process that monitors the edge of large networks and gathers netflow data on possible botnet traffic. The netflow data is then assembled and passed to the flow-correlation module. Zhang used a process developed in BotMiner called C-flow (Gu et al., 2008) to build the flow-correlation module. However, Zhang used a more efficient process for clustering netflows to allow for larger traffic volumes and employed correlation to identify hosts that had similar persistent communications. In Zhang's final process, a fine-grained detector utilizes previous detection techniques based on deep packet inspection. Zhang used both BotMiner and BotSniffer to inspect the traffic identified as possible botnet traffic by the flow-capture and flow-correlation modules and was able to achieve 100% detection rate when using cross correlation of flows and the B-sampling algorithm. For sampling rates above 0.05%, Zhang obtained false positive rates between 0.3% and 8%, as the sampling rate increased. However, when Zhang used both flow-correlation and a fine-grain detector, Zhang was able to detect 100% of botnets with no false positives for sampling rates above 0.05%.

Ilavarasan and Muthumanickam (2012) combined host level detection and network level analysis to overcome the limitations of each separately. The host level detection utilized registry analysis and file monitoring to detect changes related malware associated with botnets (Ilavarasan & Muthumanickam, 2012). Ilavarasan and Muthumanickam analyzed network traffic to identify peer-to-peer traffic and cluster similar traffic based on activity and contacted IP addresses. The final process in Ilavarasan's and Muthumanickam's detection technique was a correlation engine that combined the network analysis with the host level detection to alert for possible botnet infections.

Zeng (2012) developed a three-pronged approach to identify and mitigate the effects of botnets. Zeng proposed utilizing end host containment of infected bots, network edge detection of botnets, and measuring of network components at the infrastructure level for large botnet detection. Zeng also presented a proof of concept for future botnets utilizing mobile smart phones and SMS messages for command and control of a botnet. Zeng discussed the history of botnets and botnet detection techniques and highlighted the limitations of the current strategies to detect botnets. Most notably, the researcher discussed the rapidly changing communication methods for botnets, including peer-to-peer communications, and the limitations of current HTTP and IRC detection techniques (Zeng, 2012).

Zeng's (2012) research on end-host botnet detection incorporated previous techniques for containment of fast spreading network worms with new behavior analysis of all applications on the computer. The behavior analysis examined the actions of applications at the registry, file system and network stack, and was successful at identifying suspicious actions, while allowing legitimate applications (Zeng, 2012). Furthermore, the rate of false positives was greatly reduced when compared to existing detection techniques (Zeng, 2012)

Zeng (2012) also combined the edge network detection technique with the host-based detection to increase the effectiveness of

botnet detection. The edge network detection utilizes NetFlow data captured from routers and does not access the packet payload, ensuring privacy for legitimate traffic (Zeng, 2012). Zeng identified 17 traits of botnets that he used to determine if network traffic was suspicious and related to botnets. The 17 traits identified by Zeng for botnet traffic include the following network flow features: mean, variance, skewness, and kurtosis for duration; mean, variance, skewness, and kurtosis for total bytes; mean, variance, skewness, and kurtosis for number of packets; and the number of TCP flows, UDP flows, SMTP flows, unique IPs contacted, and number of suspicious ports.

The final portion of Zeng's (2012) technique was botnet detection at the infrastructure level. Zeng chose to focus on large peer-to-peer botnets and evaluate the feasibility of detecting peer-to-peer botnets at the Internet infrastructure level. Zeng concluded that host-based techniques for botnet detection are not reliable and network edge detection is necessary to detect botnets. Furthermore, the behavior analysis and NetFlow analysis Zeng developed is independent of the type of botnet and command and control communication a botnet utilizes, thus it greatly increases the chances of botnet detection.

Bilge, Balzarotti, Robertson, Kirda, and Kruegel (2012) developed Disclosure, a botnet detection system to detect command and control servers, rather than individual bots. Using Netflow data, Disclosure distinguishes between botnet command and control server traffic, and benign server traffic through the flow size between a server and client. Bilge et al. stated command and control server traffic does not fluctuate significantly due to the limited number of commands used by the botnet. Furthermore, the objective of the botnet is to stay undetected, so the botnet sends the shortest flow of data possible. Similar to anomaly based IDS, Disclosure performed better when larger amounts of benign flow data were analyzed. This enabled Disclosure to distinguish between benign server traffic and command and control server traffic (Bilge et al., 2012). During evaluation, Bilge et al. tried numerous settings within Disclosure and the results showed as the detection rate increased, so did the false positive rate.

Using the different behaviors of botnets, Li, Xie, Luo and Zhu (2013) developed Snort rules to detect botnet activity. Specifically, Li, Xie, et al. determined there were six behaviors unique to botnets: abnormal access to backup DNS servers, large number of domain name requests to a single domain, accessing fast flux networks, downloading malware, ingress and egress scanning, and null TCP connections. Based on these behaviors, Li, Xie et al. developed Snort rules to detect each behavior. Every Snort alert was tracked in an alert matrix and correlated against the six known botnet behaviors to identify botnet activity (Li, Xie et al., 2013). Li, Xie et al. were successful at identifying 20 known botnets with detection rates between 74% and 94% with no false positives. Lie et al. also test the Snort (Cisco, 2014) rules against 8 unknown botnets and detected between 56% and 73% of unknown botnets with zero false positives. Lie, Xie et al. explain unknown botnets as botnets that the malicious behavior of the botnet is unknown, not the actual malware.

Rossow and Dietrich (2013) recognized that existing intrusion detection systems are not capable of detecting all encrypted command and control traffic based on payload signatures. The payload-based signatures used by intrusion detection systems are easily defeated by encrypted or obfuscated command and control traffic because botnets employ defense measures against payload signature recognition, such as dynamic encryption keys,

data payloads encrypted with the XOR cipher, and varying the length of messages (Rossow & Dietrich, 2013). To counter the defenses employed by botnets, Rossow and Dietrich developed Provex, a Network Intrusion Detection system (NIDS), which detects encrypted botnet communications and was designed to learn from previously decrypted botnet communications and identify characteristic bytes within encrypted traffic. Then Provex "derives probabilistic vectorized signatures that can be used to verify if decrypted packets stem from a certain malware family's C&C" (Rossow & Dietrich, 2013, p. 6). Although Provex must decrypt network traffic and match signatures to the decrypted packets, Rossow and Dietrich were able to operate Provex at nearly 1Gbit/s of network traffic without packet loss and believed that Provex would handle network speeds of up to 10Gbit/s. In laboratory testing, Provex detected all true positive encrypted communications 100% of the time for six botnet variants and 78%, 81.5%, 87%, and 97% for four botnets, with only three false positive results (Rossow & Dietrich, 2013).

Using 1317 distinct malware samples from 8 malware families that communicate via P2P, Kheir and Wolley (2013) developed a malware classifier as part of their botnet detection technique. Kheir and Wolley recognized that P2P botnet traffic can be distinguished by three characteristics, time, space, and flow size. Using these characteristics, Kheir and Wolley used machine learning to differentiate P2P botnet traffic from benign P2P traffic. There testing showed P2P botnet traffic can be distinguished from benign P2P traffic with low false positive rates.

Garant and Lu (2013) reviewed existing botnet detection techniques and determined such were ineffective against unknown botnets as well as botnets that employ encrypted communications. Grant and Lu developed the Weasel botnet that employs fully encrypted communications to test a new detection technique that is capable of detecting encrypted botnet communications. Garant and Lu and identified six features to identify the encrypted botnet communications: length in bytes, packet count, protocol, flow duration, flow direction, and TCP flags. To develop the signature of botnet communications utilizing the six features, Garant and Lu used a decision tree classification with the C4.5 and Weka's J48 algorithms; the researchers successfully detected over 90% of encrypted botnet communications with a false positive rate of 9.9% and false negative rate of 10.5%.

Zhang, Perdisci, Lee, Luo, & Sarfraz (2014) built upon their previous work in 2011 to increase efficiency, reduce storage costs, and boost the system scalability. Zhang et al. eliminated the analysis of failed network connections for P2P traffic as an indicator of P2P botnet traffic and relied entirely on netflow analysis for botnet detection. Through hierarchical clustering of P2P flows, Zhang et al. were able to distinguish legitimate P2P traffic from botnet P2P traffic with 100% true positive detection rate and 0.2% false positive detection rate.

Using machine learning, Haddadi, Morgan, Filho, & Zincir-Heywood (2014) developed a botnet detection technique for HTTP botnets. Haddadi et al. used C4.5 and Naïve Bayes machine learning classifiers to analyze netflow data and detect HTTP botnet traffic. Since the detection technique only relies on netflow data, the technique is not affected by encrypted botnet traffic. The detection technique was tested against the Zeus botnet and the Citadel botnet. Haddadi et al. tested the detection technique with netflows containing all captured traffic and with filtered netflows of only HTTP traffic. The detection results with all traffic ranged between 7% and 88% for true positive detections and 1% to 16% for false positive

detections. When the HTTP filter was applied, Haddadi et al. increased the true positive detection rate to 85% and 97% for Zeus and Citadel traffic respectively. Furthermore, the false positive detection rates decreased to 14% and 3%.

4. COMPARISON OF BOTNET DETECTION TECHNIQUES BY INFRASTRUCTURE

This section provides a comparison of botnet detection techniques. We have compared the techniques based on the techniques ability to detect IRC, HTTP, and P2P based botnets and whether the technique is effective against encrypted botnet communications. Table 2 provides a summary of the different techniques detection ability.

Table 2 also shows the change in detection techniques as botnets changed communication methods and infrastructures. Between 2005 and 2007 researchers focused IRC and HTTP botnets that use a centralized command and control server. Then in 2008 detection techniques began to include P2P communication and decentralized infrastructures. Finally, in 2012 detection techniques began to include the ability to detect encrypted communications.

Previous botnet detection techniques have reported varied success rates for botnet detection and rates of false positive detections. In 2007 and 2008 true positive detection rates ranged between 95% and 96.8%, while false positive detection rates were between 0.049% and 0.0003%. (Gu et al., 2007; Gu, Perdisci, et al., 2008). Between 2009 and 2014 true positive detection rates increased to between 99% and 100%, however false positive detection rates also increased to a range of 0.0056% and 0.2% (Barthakur et al., 2012; Francois et al., 2011; Haddadi et al., 2014; Wang and Yu, 2009; Zeng et al., 2010; Zhang et al., 2011, Zhang et al., 2014). This survey showed that while true positive detection rates have increased, so have false positive detection rates. The one exception to these results are from Haddadi et al. (2014), were the true positive detection rates decreased. Table 3 shows the true detection rates and false positive detection rates for nine studies reviewed as part of this survey that provided detection rates.

Table 2
Detection Capabilities of Different Botnet Detection Techniques

Researchers	IRC	HTTP	P2P	Encrypted
Cooke, Jahanian and McPherson (2005)	X			
Gu, Porras and Yegneswaran (2007)	X			
Karasaridis, Rexford, and Hoeflin (2007)	X			
Gu, Zhang, and Lee (2008)	X	X		
Gu, Perdisci et al. (2008)	X	X	X	
Wang and Yu (2009)	X	X		
Nagaraja, Mittal, Hong, Caesar and Borisov (2010)			X	
Zeng, Hu and Shin (2010)	X	X	X	
Francois, Wang, Bronzi, State and Engel (2011)			X	
Zhang, Perdisci, Lee, Sarfraz and Luo (2011)			X	
Barthakur, Dahal and Ghose (2012)		X	X	X
Han, Chen, Xu and Liang (2012)	X	X		
Zhang (2012)	X	X	X	
Ilavarasan and Muthumanickam (2012)			X	
Zeng (2012)	X	X	X	X
Li, Xie, Luo and Zhu (2013)	X	X	X	X
Rossow and Dietrich (2013)	X	X	X	X
Garant and Lu (2013)		X		X
Zhang et al. (2014)			X	X
Haddadi et al. (2014)		X		X

Table 3
Botnet Detection Rates

Researchers	True Positive Rate	False Positive Rate
Gu et al. (2007)	95.1%	0.049%
Gu, Perdisci et al. (2008)	96.83%	0.0003%
Wang and Yu (2009)	100%	0.0056%
Zeng et al. (2010)	99.99%	0.16%
Francois et al. (2011)	99%	0.1%
Zhang et al. (2011)	100%	0.2%
Barthakur et al. (2012)	99.01%	0.11%
Zhang et al. (2014)	100%	0.2%
Haddadi et al. (2014) Citadel	97%	3%
Haddadi et al. (2014) Zeus	85%	14%

Note: Gu, Perdisci et al. (2008) true detection rate is an average of 8 tests

An analysis of the botnet detection techniques reviewed in the survey showed that techniques which used machine learning and hierarchical clustering of flow data were more effective than techniques based on deep packet analysis or fingerprint analysis. The same was true for the efficiency and scalability of the techniques. Relying solely on Netflow data allows the techniques to process large data sets, while maintaining high true positive detection rates and low false positive rates.

5. CONCLUSION

This survey examined the existing research on botnet detection and distributed denial of service attacks in a chronological order. Literature was reviewed from numerous sources including scholarly journals, conference papers, books, dissertations, and government documents. The literature was obtained from numerous online databases including, ProQuest, IEEE Computer Society Digital Library, ACM Digital Library, Google Scholar, and the IEEE Xplore Digital Library. The keywords used in the search included botnet, distributed denial of service, malware, denial of service, botnet detection, botnet identification, and proactive botnet. The review showed that botnets and botnet detection techniques are constantly evolving as Botmasters update and modify botnets to stay ahead of the latest botnet detection techniques (Alhomoud et al., 2013; Garant & Lu, 2013; Zargar, Joshi, & Tipper, 2013). Although IRC and HTTP botnets are still active, most new botnets use a decentralized infrastructure to avoid a single point of failure (Garant & Lu, 2013; Gu et al., 2009). Furthermore, a majority of botnets now utilize encrypted communications to avoid detection (Garant & Lu, 2013; Gu et al., 2009; Li, Xie et al., 2013; Rossow & Dietrich, 2013). Therefore, modern botnet detection techniques attempt to detect botnet command and control communications within network traffic through hierarchical clustering of flow data (Haddadi et al., 2014; Kheir & Wolley, 2013; Zhang et al., 2014).

REFERENCES

Alhomoud, A., Awan, I., Disso, J., & Younas, M. (2013). A next-generation approach to combating botnets. Computer, 46(4), 62-66. Retrieved from http://doi.ieeecomputersociety.org/10.1109/MC.2013.67

Bailey, M., Cooke, E., Jahanian, F., Yunjing, X., & Karir, M. (2009). A survey of botnet technology and defenses. *Proceedings of the 2009 Conference for Homeland Security*, Washington, DC, 299-304. Retrieved from http://dx.doi.org/10.1109/CATCH.2009.40

Bilge, L., Balzarotti, D., Robertson, W., Kirda, E., & Kruegel, C. (2012, December). Disclosure: Detecting botnet command and control servers through large-scale Netflow analysis. *Proceedings of the 28th Annual Computer Security Applications Conference*, New York, NY, 129-138. Retrieved from http://dl.acm.org/citation.cfm?id=2420969

Brezo, F., Santos, I., Bringas, P., & Val, J. (2011, Aug). Challenges and limitations in current botnet detection. *Proceedings of the 22nd International Workshop on Database and Expert Systems Applications*, Toulouse, France, 95-101. Retrieved from http://dx.doi.org/10.1109/DEXA.2011.19

Caglayan, A., Toothaker, M., Drapaeau, D., & Burke, D. (2010, January). Behavioral patterns of fast flux service networks. *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences (HICSS),* Honolulu, HI, 1-9. doi: 10.1109/HICSS.2010.81

Cao, L, & Qiu, X. (2013, July). Defense against botnets: A formal definition and a general framework. *Proceedings of the 2013 IEEE Eighth International Conference on Networking, Architecture, and Storage*, Xi'an, Shaanxi, China, 237-241. Retrieved from http://doi.ieeecomputersociety.org/10.1109/NAS.2013.37

Cisco. (2014). Snort (Version 2.9.6.2) [Computer Software]. Retrieved from http://www.snort.org/downloads

Cooke, E., Jahanian, F., & McPherson, D. (2005, July). The zombie roundup: Understanding, detecting, and disrupting botnets. *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet Workshop 2005,* Cambridge, MA. Retrieved from https://www.usenix.org/legacy/events/sruti05/tech/full_papers/cooke/cooke.pdf

Dean, J., & Ghemawat, S. (2004, December). MapReduce: Simplified data processing on large clusters. *Proceedings of the 6th Symposium on Operating System Design and Implementation*, San Francisco, CA, 137-150. Retrieved from http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/archive/mapreduce-osdi04.pdf

Dittrich, D. (2012, April). So you want to take over a botnet. *Proceedings of the 5th USENIX Workshop on Large-Scale Exploits and Emergent Threats, LEET '12*, San Jose, CA. Retrieved from https://www.usenix.org/system/files/conference/leet12/leet12-final23.pdf

Feily, M., Shahrestani, A., & Ramadass, S. (2009, June). A survey of botnet and

botnet detection. *Proceedings of the 2009 Third International Conference on Emerging Security Information, Systems and Technologies*, Athens, Glyfada, Greece, 268-273. Retrieved from http://doi.ieeecomputersociety.org/10.1109/SECURWARE.2009.48

Francois, J., Wang, S., Bronzi, W., State, R., & Engel, T. (2011, November). BotCloud: Detecting botnets using Mapreduce. *Proceedings of the 2011 IEEE International Workshop on Information Forensics and Security*, Iguazu Falls, Parana, Brazil, 1-6. Retrieved from http://dx.doi.org/10.1109/WIFS.2011.6123125

Francois, J., Wang, S., State, R., & Engel, T. (2011). BotTrack: tracking botnets using NetFlow and PageRank. *Proceedings of the 10th International IFIP TC 6 Conference on Networking,* Heidelbert, Germany, 1-14. Retrieved from http://dl.acm.org/citation.cfm?id=2008782

Garant, D., & Lu, Wei. (2013). Mining botnet behaviors on the large-sale web application community. Proceedings of the *2013 27th International Conference on Advanced Information Networking and Applications Workshops*, Barcelona, Spain, 185-190. Retrieved from http://doi.ieeecomputersociety.org/10.1109/WAINA.2013.235

Gu, G., Perdisci, R., Zhang, J., & Lee, W. (2008, July). BotMiner: Clustering analysis of network traffic for protocol and structure independent botnet detection. *Proceedings of the 17th USENEX Security Symposium,* San Jose, CA. Retrieved from https://www.usenix.org/legacy/event/sec08/tech/full_papers/gu/gu.pdf

Gu, G., Porras, P., Yegneswaran, V., Fong, M., & Lee, W. (2007, August). BotHunter: Detecting malware infection through IDS-driven dialog correlation. *Proceedings of the 16th USENEX Security Symposium,* Boston, MA. Retrieved from https://www.usenix.org/legacy/events/sec07/tech/full_papers/gu/gu.pdf

Gu, G., Yegneswaran, V., Porras, P., Stoll, J., & Lee, W. (2009, December). Active botnet probing to identify obscure command and control channels. *Proceedings of the 2009 Annual Computer Security Applications Conference,* Honolulu, HI, 241-253. doi: 10.1109/ACSAC.2009.30

Gu, G., Zhang, J., & Lee, W. (2008, February). BotSinffer: Detecting botnet command and control channels in network traffic. *Proceedings of the 15th Annual Network and Distributed System Security Symposium,* San Diego, CA. Retrieved from http://www.isoc.org/isoc/conferences/ndss/08/papers/17_botsniffer_detecting_botnet.pdf

Haddadi, F., Morgan, J., Filho, E., & Zincir-Heywood, A. (2014). Botnet behaviour analysis using IP flows: With HTTP filters using classifiers. *Proceedings of the 28th International Conference on Advanced Information Networking and Applications Workshops*, Victoria, British Columbia, 7-12. Retrieved from http://dx.doi.org/10.1109/WAINA.2014.19

Hadoop. (2013). The Apache Hadoop project. Retrieved from http://hadoop.apache.org/

Han, F., Chen, Z., Xu, H., & Liang, Y. (2012, June). Garlic: A distributed botnets suppression system. *Proceedings*

of the 2012 32nd International Conference on Distributed Computing Systems Workshops, Macau, China, 634-639. Retrieved from http://doi.ieeecomputersociety.org/10.1109/ICDCSW.2012.30

Hasan, A., Awadi, R., & Belaton, B. (2013). Multi-phase IRC botnet and botnet behavior detection model. *International Journal of Computer Applications, 66*(15), 41-51. doi: 10.5120/11164-6289

Householder, A., & Danyliw, R. (2003, March). *Increased activity targeting windows shares* (CERT advisory CA-2003-08). Retrieved from http://www.cert.org/advisories/CA-2003-08.html

Karasaridis, A., Rexford, B., & Hoeflin, D. (2007, April). Wide-scale botnet detection and characterization. *Proceedings of the First Workshop on Hot Topics in Understanding Botnets*, Cambridge, MA. Retrieved from https://www.usenix.org/legacy/event/hotbots07/tech/full_papers/karasaridis/karasaridis.pdf

Kheir, N., & Wolley, C. (2013). BotSuer: Suing stealthy P2P bots in network traffic through Netflow analysis. In M. Abdalla, C. Nita-Rotaru, & R. Dahab (Eds.), *Cryptology and Network Security* (pp. 162-178). doi: 10.1007/978-3-319-02937-5_9

Li, W., Xie, S., Luo, J., & Zhu, X. (2013, April). A detection method for botnet based on behavior features. *Proceedings of the 2nd International Conference on Systems Engineering and Modeling (ICSEM-13)*, Beijing, China, 512-517. Retrieved from http://www.atlantis-press.com/php/download_paper.php?id=5594

Roscini, M. (2014). Cyber operations and the use of force in international law. New York, NY: Oxford University Press

Rossow, C., & Dietrich, C. (2013, July). PROVEX: Detecting botnets with encrypted command and control channels. *Proceedings of the 10th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment,* Berlin, Heidelberg, 21-40. Retrieved from http://dx.doi.org/10.1007/978-3-642-39235-1_2

Spitzner, L. (2003). The honeynet project: Trapping the hackers. *IEEE Security & Privacy, 1*(2), 15-23. doi: 10.1109/MSECP.2003.1193207

Ventre, D. (2013). Cyber Conflict: Competing National Perspectives. Indianapolis, IN: Wiley. Wang, T., & Yu, S. (2009). Centralized botnet detection by traffic aggregation. Proceedings of the 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications, Chengdu, China, 86-93. Retrieved from http://dx.doi.org/10.1109/ISPA.2009.74

Zargar, S., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (distributed denial of service) flooding attacks. *IEEE Communications Surveys and Tutorials, PP*(99), 1-24. doi: 10.1109/SURV.2013.031413.00127

Zeng, Y. (2012). *On detection of current and next-generation botnets* (Doctoral dissertation). University of Michigan. Retrieved from http://deepblue.lib.umich.edu/handle/2027.42/91382

Zeng, Y., Hu, X., & Shin, K. (2010, June). Detection of botnets using combined host

and network level information. *Proceedings of the 2010 IEEE/IFIP International Conference on Dependable Systems and Networks*, Chicago, IL, 291-300. Retrieved from http://doi.ieeecomputersociety.org/10.1109/DSN.2010.5544306

Zhang, J. (2012). *Effective and scalable botnet detection in network traffic.* (Doctoral Dissertation). Retrieved from ProQuest Dissertations and Theses database. (AAT 1115317916)

Zhang, J., Perdisci, R., Lee, W., Sarfraz, U., & Luo, X. (2011, June). Detecting stealthy P2P botnets using statistical traffic fingerprints. *Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks*, Hong Kong, China, 121-132. Retrieved from http://doi.ieeecomputersociety.org/10.1109/DSN.2011.5958212

Zhang, J., Perdisci, R., Lee, W., Luo, X., & Sarfraz, U. (2014). Building a scalable system for stealthy P2P-Botnet detection. *IEEE Transactions on Information Forensics and Security, 9*(1), 27-38. Retrieved from http://dx.doi.org/10.1109/TIFS.2013.2290197

Zhu, Z., Lu, G., Chen, Y., Fu, Z., Roberts, P., & Han, K. (2008) Botnet research survey. *Proceedings of the 32nd Annual IEEE, International Computer Software and Applications*, Turku, Finland, 967–972. Retrieved from http://dx.doi.org/10.1109/COMPSAC.2008.205

Zhuge, J., Holz, T., Han, X., Guo, J., & Zou, W. (2007, December). *Characterizing the IRC-Based Botnet Phenomenon.* Peking University and University of Mannheim Technical Report. Retrieved from https://ub-madoc.bib.uni-mannheim.de/1710/1/botnet_china_TR.pdf

