Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Review:

Botnet detection techniques: review, future trends, and issues*

Ahmad KARIM†1, Rosli Bin SALLEH[1], Muhammad SHIRAZ[1], Syed Adeel Ali SHAH[1],
Irfan AWAN[2], Nor Badrul ANUAR[1]

(*[1]Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia*)
(*[2]Department of Computer Science, University of Bradford, Bradford BD7 1DP, UK*)
†E-mail: ahmadkarim@um.edu.my

Received Aug. 31, 2013; Revision accepted Jan. 23, 2014; Crosschecked Oct. 15, 2014

Abstract: In recent years, the Internet has enabled access to widespread remote services in the distributed computing environment; however, integrity of data transmission in the distributed computing platform is hindered by a number of security issues. For instance, the botnet phenomenon is a prominent threat to Internet security, including the threat of malicious codes. The botnet phenomenon supports a wide range of criminal activities, including distributed denial of service (DDoS) attacks, click fraud, phishing, malware distribution, spam emails, and building machines for illegitimate exchange of information/materials. Therefore, it is imperative to design and develop a robust mechanism for improving the botnet detection, analysis, and removal process. Currently, botnet detection techniques have been reviewed in different ways; however, such studies are limited in scope and lack discussions on the latest botnet detection techniques. This paper presents a comprehensive review of the latest state-of-the-art techniques for botnet detection and figures out the trends of previous and current research. It provides a thematic taxonomy for the classification of botnet detection techniques and highlights the implications and critical aspects by qualitatively analyzing such techniques. Related to our comprehensive review, we highlight future directions for improving the schemes that broadly span the entire botnet detection research field and identify the persistent and prominent research challenges that remain open.

Key words: Botnet detection, Anomaly detection, Network security, Attack, Defense, Taxonomy
doi:10.1631/jzus.C1300242 **Document code:** A **CLC number:** TP393.08

1 Introduction

The latest developments in the communication and computing technologies have directed users towards distributed computing. E-email, web applications, and voice over IP applications are common examples of distributed services employed over the Internet; however, malicious software has acquired an important position in the evolving distributed computing models. Software containing malicious functionality has existed ever since the earliest use of programmable systems; however, malware often has limited or just local impact. In recent years, the success of the Internet has become a starting point for reporting widespread malware infections which affect millions of systems around the world. As a result, botnets, which are remotely controlled networks of hijacked computers, have become popular. The basic aim of these distributed coordinated networks is to initiate various malicious activities over the network including phishing, click fraud, spam generation, copyright violations, key logging, and most importantly, the denial of service (DoS) attacks. Botnets are viewed as serious threats to network resources over the Internet (Fossi *et al.*, 2011). Currently, there is an increasing competition in the botnet market (Bu *et al.*, 2010). A number of new programs were introduced in the beginning of 2010, such as Buga, Spy Eye, Clod, and Filon (Truhanov, 2010), for the implementation of botnets. Moreover, the basic approach of a botmaster is to preserve the bots for the longest time to achieve the maximum benefit.

* Project supported by University of Malaya, Malaysia (No. FP034-2012A)

Therefore, for the maximum benefit attainment, bots use different vigorous invasive approaches to hide their malicious intension. For instance, malware code can be hidden in a form that may not be detected by various signature based antivirus software. Moreover, bots use standard/common protocols, e.g., HTTP, Internet Relay Chat (IRC), and peer-to-peer (P2P), to carry out their communications and try to set activity levels below normal computer/user levels (di Pietro and Mancini, 2008).

As a result, a number of commercial, noncommercial, and government organizations have been the targets for botnet attackers. For example, Estonia in 2007 and Georgia in 2008 were out of service for several days because of DDoS attacks (Nazario, 2009). In addition, the Stuxnet botnet (Falliere *et al.*, 2011) was observed in 2009, causing cooperate intellectual property to be stolen by capturing SCADA systems. In October 2009, the FBI disclosed that its losses due to a botnet attack were valued at more than 100 million dollars. The intensity of DDoS attacks, which are considered the most dangerous attacks, is increasing with the growing use of the Internet. Therefore, botnets act as a platform for launching worms or viruses instantaneously with the help of bot (infected machine) enemies. Statistics show that botnet is becoming the curse problem of the current times. Some reports indicate that more than 80% of the Internet traffic is propagating through botnets such as Grum, Cutwail, and Rustock botnets (Mador, 2012). Although such network spam attacks can be controlled at their destination end, they can still initiate such attacks and greatly distribute spams through network backbones, which will result in the abundant utilization of network resources. It was reported that the distribution of a botnet is as cheap as distributing 10000 bots for only 15 USD (Mador, 2012). The International Telecommunication Union (ITU) reported that the malware distribution has caused a damage of 13.2 billion to 67.2 billion USD to the global market during the years 2005 to 2007 (Bauer *et al.*, 2008).

The botnet has become a most threatening phenomenon and shown its harmful effect on network communities over the last decade. Researchers, law-enforcement authorities, businesses, and individuals have started to discover methods to combat this malicious threat (Cooke *et al.*, 2005; Ceron *et al.*, 2008; Choi *et al.*, 2009). Botnet detection is currently an ongoing challenge for researchers and organizations. Botnets are considered moving targets, which means all the aspects of botnets including detection, mitigation, and response are changing over time; therefore, no mitigation or detection technique offers a permanent solution. Similarly, different types of stakeholders, for instance, enterprises, governments, networks, and Internet service providers (ISPs), have different ways and goals to address the issue of botnets. Moreover, with the advent of new technologies and increase in the knowledge base, the expertise of botmasters is improving in evading botnet detection techniques and trying to rally sophistication for the command and control (C&C) architecture.

Currently, a number of review articles (Paxton *et al.*, 2007; Bailey *et al.*, 2009; Feily *et al.*, 2009; Jing *et al.*, 2009; Li *et al.*, 2009; Shin and Im, 2009; Zeidanloo and Manaf, 2009; Marupally and Paruchuri, 2011; Plohmann *et al.*, 2011; Silva *et al.*, 2013; Rodríguez-Gómez *et al.*, 2013) are available, which cover only the botnet techniques before 2009 and therefore lack analysis of the latest trends in the state-of-the-art for the botnet detection phenomenon. Feily *et al.* (2009) classified the botnet phenomenon and categorized botnet detection techniques into four broad categories: signature-, DNS-, mining-, and anomaly-based. Moreover, they provided a comparison chart to highlight the importance of these techniques with respect to their detection approaches such as unknown bot detection, protocol and structure independence, encrypted detection, real-time detection, and low false positive rate. Similarly, Bailey *et al.* (2009) and Shin and Im (2009) primarily focused on the botnet environment in general and characterized botnets based on propagation mechanisms (OS, services, applications, and social engineering), C&C topologies (centralized C&C, P2P, unstructured), and different attack classes (DDoS, identity theft, spam, phishing). Moreover, Bailey *et al.* (2009) classified botnet detection techniques into different approaches, such as detection based on corporate behavior, signatures, and attack behavior. Li *et al.* (2009) discussed the botnet and its related research based on C&C models, infection mechanism, communication protocols, malicious behavior, and defensive mechanisms. Jing *et al.* (2009) presented the basic architecture of IRC based botnet attacks, wherein malicious activities were detected by directly monitoring IRC communication patterns. This scheme correlates common traffic patterns along with the additional features. Common traffic patterns that do not relate to the human standards are considered bots in the network. The review articles by Paxton *et al.* (2007), Zeidanloo and Manaf (2009), Marupally and Paruchuri (2011), and Plohmann *et al.* (2011) discussed different C&C architectures (e.g., centralized, P2P, and hybrid) for botnets. In addition to that, Zeidanloo *et al.* (2010) classified botnet detection techniques into honeynets and intrusion detection systems (IDSs), which were then further characterized as anomaly- and signature-based detection methods.

A more recent survey (Silva *et al.*, 2013) highlights the main lines of research of the botnet phenomenon in general, including botnet architecture, life cycle, creation, detection, and mitigation approaches. Moreover, a comprehensive study and taxonomy was derived through exploration of different botnet life cycle stages (Rodríguez-Gómez *et al.*, 2013) and it is depicted from the findings that, every stage of the botnet life cycle is completed solely to drive the success of the whole botnet. Therefore, any interruption during the execution of just one stage (recruitment, interaction, marketing, or attack execution) in the botnet life cycle may lead to the complete botnet being useless.

Apart from the above mentioned review articles, a deeper and wider study is needed to analyze the recent algorithms developed for botnet detection. This paper presents a comprehensive review of the latest state-of-the-art techniques for botnet detection to demonstrate the main ideas associated with previous and current research. We propose a thematic taxonomy for the classification of botnet detection techniques and investigate the implications and critical aspects of such techniques by qualitative analysis. Furthermore, we identify the persistent and prominent research challenges that remain open and highlight future directions for enhancing and improving the schemes that broadly span the entire botnet detection research domain. Therefore, our contribution is different from previous studies in that this article is focusing mainly on the botnet detection domain, keeping in mind the latest trends involving the categorization of botnet detection techniques based on thematic taxonomy, analysis of current techniques by discussing the implications and critical aspects, accompanied with identification of the open issues and challenges. The listing of challenges and open issues guides researchers to select the appropriate domain for future research and obtain ideas to further explore the botnet phenomenon.

The rest of this paper is organized as follows. Section 2 describes the fundamental concepts of the botnet phenomenon, including botnet timeline, botnet life cycle, and botnet architecture. Section 3 presents the taxonomy for the classification of botnet detection techniques, focusing on latest botnet detection trends. Section 4 focuses on recent trends for the botnet detection phenomenon. Section 5 highlights the open challenges concerning botnet detection. Section 6 concludes this paper and draws attention to the future direction for ongoing research.

2 Background

This section discusses the theoretical framework of the botnet phenomenon. It describes the fundamental concepts of the botnet phenomenon including a botnet timeline, botnet life cycle, and botnet architecture.

2.1 Botnet phenomenon

Botnet has become a threatening phenomenon for the dissemination of various Internet attacks including spamming, distributed denial of service (DDoS) attacks, and malicious activities. Botnet is a network of infected machines (also called ‘bots’) which aims to disseminate malicious code over the Internet without user intervention. This process is carried out by a centralized entity called ‘C&C’, which is also called a ‘botmaster’. Therefore, the theme of C&C mechanism is to increase the number of bot enemies and to coordinate among those enemies for the intensive destructive operations which are then carried out. The difference between a botnet and other types of network attacks is the existence of C&C. In addition, the infected machines (bots) receive instructions from C&C and act upon those instructions. The instructions/commands range from initiating a worm or spam attack over the Internet to disrupting a legitimate user request. Bots are computer machines with malicious software installed on them, and they interact with an individual's machine without being noticed or even without any intervention by the user. A botmaster is the entity to whom all infected machines (bots) coordinate to initiate, manage, or suspend attacks. A botnet causes a number of serious offences on the Internet, as it allows intruders to hijack several computers simultaneously, which increases the number of cyber-attacks (Paxton *et al.*, 2007). The research on the botnet is evolving rapidly because of the increasing curiosity in the Internet community. An exploration of many scientific databases including IEEE, ACM, Elsevier, and Science Direct reveals the increasing number of articles published from 2005 to April, 2013 on the botnet (Fig. 1).

The concept of 'botnet' evolved in 1993 by introducing the first botnet called 'Eggdrop' (Wang, 2003). The history of botnets is highlighted in Table 1. The year field shows the commencement year of each botnet, the 'number of estimated bots' refers to the number of bots anticipated in the given botnet attack, 'spam capacity' shows the number of attacks (per day) that hinders the services of legitimate users. Similarly, 'aliases' refers to the different naming conventions used by each botnet. Moreover, botnet detection and mitigation approaches used in response to these attacks are listed, and most importantly, the type of attack (IRC, P2P, SMTP, HTTP, etc.).

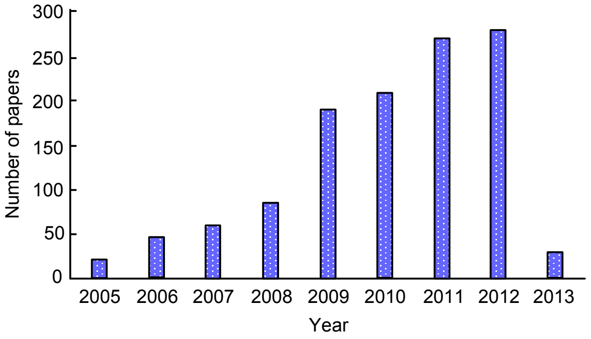

Fig. 1 Number of publications on the botnet from 2005 to April 2013

2.2 Botnet life cycle

Fig. 2 shows the general flow chart/life cycle for the botnet phenomenon. To become an active bot for a host machine, it follows specified steps (Zhu *et al.*, 2008; Feily *et al.*, 2009; Schiller *et al.*, 2011).

Table 1 Botnet timeline

Year	Name	Number of estimated bots	Spam capacity (billion/d)	Aliases	Detection approach	Type	Reference
1993	Eggdrop	-	-	Valis	-	IRC	Wang (2003)
1998	GTBot	-	-	Aristotles	-	mIRC	Janssen (2011)
	NetBus	-	-	NetPrank	AV software	HTTP	Wikipedia (1998)
1999	!A	1 billion	-	-	-	-	Wikipedia (2013b)
2002	Sdbot/Rbbot	-	-	IRC-SDBot	Data mining, SVM	IRC	Sevcenco (2012)
	Agobot	-	-	W32.HLLW.Gaobot, Gaobot	Expert system	IRC	Podrezov (2013)
2003	Spybot	-	-	-	-	P2P, IRC	Schiller and Binkley (2007)
	Sinit	-	-	Win32.Sinit, Troj/BDSinit	Network flow analysis	P2P	Wang *et al.* (2007)
2004	Bobax	100 000	27	-	-	-	Kassner (2003)
	Bagle	230 000	5.7	Beagle, Mitglieder	Symantec	SMTP	Symantic (2010)
2006	Rustock	150 000	30	RKRustok, Costrat	Operation b107	IRC	Miller (2008)
2007	Akbot	1 300 000	-	-	Operation: bot roast	IRC	The H Security (2007)
	Cutwail	1 500 000	74	Pandex, Mutant	-	SMTP	Marry (2010)
	Srizbi	450 000	60	Cbeplay, Exchanger	Symantec	IRC	BBC (2008)
	Storm	160 000	3	Nuwar, Peacomm, Zhelatin	Fast flux	P2P	Francia (2007)

To be continued

Table 1

Year	Name	Number of estimated bots	Spam capacity (billion/d)	Aliases	Detection approach	Type	Reference
2008	Conficker	10 500 000+	10	DownAndUp, Kido	AV software	HTTP/P2P	Schmudlach (2009)
	Mariposa	12 000 000	-	-	Manual	IRC/HTTP	McMillan (2010)
	Sality	1 000 000	-	Sector, Kuku, Kookoo	Manual	P2P	Falliere (2011)
	Asprox	15 000	-	Danmec, Hydraflux	Symantec	HTTP	Goodin (2008)
	Gumblar	n/a	-	-	Manual	HTTP	Mills (2009)
	Waledac	80 000	1.5	Waled, Waledpak	Kaspersky	SMTP/P2P	Goodin (2010)
	Onewordsub	40 000	1.8	N/A	-	SMTP	Keizer (2008)
	Xarvester	10 000	0.15	Rlsloup, Pixoliz	McAfee	SMTP	Symantic (2010)
	Mega-D	509 000	10	Ozdok	Manual	HTTP	Warner (2010)
	Torpig	180 000	-	Sinowal, Anserin	ESET	HTTP/IRC	Miller (2009)
	Bobax	185 000	9	Bobic, Oderoor, Cotmonger	Manual/ BitDefender	HTTP	Symantic (2010)
	Lethic	260 000	2	None	Symantec	IRC	Symantic (2010)
	Kraken	495 000	9	Kracken	Scan IP addresses	IRC	Jackson (2008)
2009	Maazben	50 000	0.5	-	-	SMTP	Symantic (2010)
	Grum	560 000	39.9	Tedroo	FireEye researchers	SMTP	Danchev (2009)
	Festi	n/a	2.25	Spamnost	ESET	SMTP/DoS	Morrison (2012)
	BredoLab	30 000 000	3.6	Oficla	Symantec	HTTP/SMTP	Crowfoot (2012)
	Donbot	125 000	0.8	Buzus, Bachsoy	Symantec	HTTP	Stewart (2009)
	Wopla	20 000	0.6	Pokier, Slogger, Cryptic	Manual/PC tools	HTTP	Keizer (2008)
	Zeus	3 600 000	n/a	Zbot, PRG, Wsnpoem	-	-	Messmer (2009)
2010	Kelihos	300 000+	4	Hlux	Kaspersky	P2P	Stefan (2013)
	TDL4	4 500 000	n/a	TDSS, Alureon	Kaspersky's TDSS killer	IRC	Kespersky (2011)
	LowSec	11 000+	0.5	LowSecurity, FreeMoney	Symantec	HTTP	Symantic (2010)
	Gheg	30 000	0.24	Tofsee, Mondera	Manual	DoS	Symantic (2010)
2011	Flashback	600 000	n/a	BacDoor.Flashback.39	Java program	P2P	Musil (2012)
2012	Chameleon	120 000	-	-	-	HTTP	Spider (2013)
2013	Boatnet	500+ server computers	0.01	YOLOBotnet			Wikipedia (2013b)

The 'initialization' is the preliminary step in the botnet life cycle, through which the botmaster sets up bot parameters to start communication. After the initial stage there is a registration process, which takes place between the botmaster and the distributed domain name system (DDNS) which assigns a static IP address to the botmaster. At the preliminary injection stage, the regular infection procedure is carried out in various forms, for example, virus propagation, through unwanted downloads, by downloading and running malicious attachments from emails, or by infected removable disk drives (Abu Rajab *et al.*, 2006; Grizzard *et al.*, 2007; Zhu *et al.*, 2008; Barsamian, 2009; Feily *et al.*, 2009). Bots build their networks in the proceeding stage and install malicious code. The infected machine performs searches, and malware binaries are installed from the database located on the network. After downloading these malicious binaries into the system, the host acts as a real bot. Moreover, the downloading process usually occurs through HTTP, FTP, or P2P protocols.

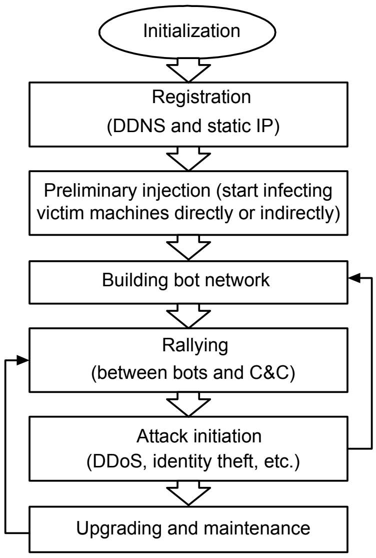
Initialization
Registration
(DDNS and static IP)
Preliminary injection (start infecting victim machines directly or indirectly)
Building bot network
Rallying
(between bots and C&C)
Attack initiation
(DDoS, identity theft, etc.)
Upgrading and maintenance

Fig. 2 Botnet flow diagram

The next step is called 'rallying', in which the connection is built up between the bots and their C&C; some researchers named it the 'connection phase' (Feily *et al.*, 2009). In fact, this step always takes place whenever a bot restarts and ensures the connection establishment status between the botmaster and bots, so that the bots take part in the botnet process and can receive commands for taking actions. Therefore, the rallying stage is a continuous process in the whole bot life cycle (Liu *et al.*, 2008). After the successful establishment of the C&C channel with the bots, in the attack stage, the bots wait for commands from their C&C and start malicious activities as prescribed by C&C. The ultimate aim of the botnets is to perform malicious activities including DDoS attacks, analyzing network traffic, pilfering computer/network resources, propagating malware on networks, searching for loopholes and vulnerabilities in the computer systems, identity theft, exploiting private documents, and manipulating games and surveys (Puri, 2003; Ianelli and Hackworth, 2005; Trend Micro, 2006; Zhu *et al.*, 2008; Zeidanloo and Manaf, 2009; Zeidanloo *et al.*, 2010). The last stage of the botnet life cycle is related to upgradation and maintenance of the malware. Maintenance is a required step that keeps the botmasters with their army of bots up to date for further coordinated attacks. Moreover, there are many reasons for updating the binary code for the bot army, such as evading different detection techniques, avoiding similar behavior, and adding new feature sets for connection with various C&C channels (Zhu *et al.*, 2008; Barsamian, 2009; Feily *et al.*, 2009). This phase is usually considered a vulnerable step, as the botnet may be detected in this stage by observing similar network behavior; therefore, it is the sole responsibility of the botmaster to make sure that the changes are reflected as quickly as possible.

2.3 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 have been proposed (Trend Micro, 2006; Gu *et al.*, 2008b; Liu *et al.*, 2008; Jing *et al.*, 2009; Wang B *et al.*, 2010), including Internet Relay Chat (IRC) centralized architecture (Kalt, 2000) and decentralized P2P architecture (Jing *et al.*, 2009) which were recently extended to HTTPS and Twitter based networks. Fig. 3 shows the taxonomy of the botnet architectures.

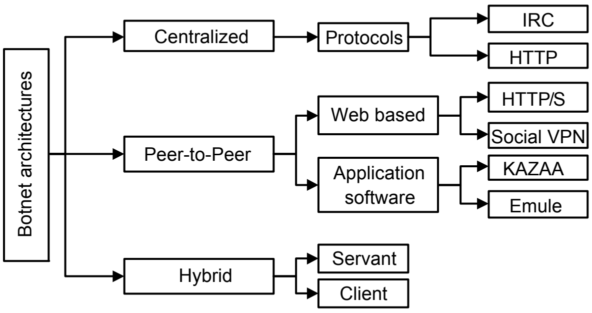

Fig. 3 Taxonomy of botnet architectures

Centralized C&C architecture: The centralized C&C approach resembles the traditional client/server architecture. The IRC protocol (Trend Micro, 2006) 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 responsibility of sending commands to bots and delivering malware updates takes place. IRC (Kalt, 2000) and the HyperText Transfer Protocol (HTTP) (McMillan, 2009) are considered the main protocols in centralized architecture. The advantages of centralized architecture include: (1) easy deployment, as it does not require any specialized hardware; (2) quick response, as the server is directly coordinating with its bots without being intervened by a third party; (3) better coordination with the bot enemy; (4) good accessibility, as there is direct coordination between the botmaster and the bots; (5) timely updates from the botmaster; (6) good scalability.

According to Stephens (2010), centralized C&C architecture can be further classified into IRC and HTTP based approaches. The drawback of a centralized approach is that, the C&C server is considered a single point of failure (Trend Micro, 2006; Wang B *et al.*, 2010), so it is quite easy to turn off detected botnets. Therefore, decentralized C&C architecture is employed to overcome the drawback of C&C architecture for the botnet phenomenon.

Decentralized/peer-to-peer (P2P) architecture: In a decentralized architecture (Jing *et al.*, 2009) modern botnets take greater flexibility to acquire a larger number of bots and to achieve the maximum benefit/profit. It is difficult to avoid decentralized botnets for the following reasons: (1) The dismissal of a whole botnet depends upon the discovery of several bots operating under a single botnet; therefore, it is difficult to capture several bots in a P2P architecture, which ultimately leads to the destruction of the whole botnet. (2) The P2P botnet lacks a centralized C&C network; therefore, it is difficult to diagnose the total area affected by any botnet. (3) It is difficult to suspend a P2P botnet because of the loosely coupled interdependence between bots. The P2P architecture is classified according to applications related to web-based applications and desktop application software (Dagon *et al.*, 2007). The newly emerging P2P web-based model (Cai and Zou, 2012) differs from old client/server models as it provides a decentralized and distributed architecture for the botmaster to more broadly disseminate criminal activities. Similarly, this distributed architecture is desirable because it avoids a single point of failure (SPOF). Attackers use HTTPS ports to hide encrypted malicious code from external firewalls or filters (Paranoid, 2004).

A more recent, free, and open-source web-based P2P application called ‘social VPN’ (WordPress, 2008) has emerged to allow directly connecting computers in a shared community. Social VPN is considered the replacement of existing applications such as Twitter and Facebook. The advantages provided by social VPN include encrypted and authenticated communication, eXtensible Messaging and Presence Protocol (XMPP) supported backbends such as jabber.org and Google Chat, and seamless access to the remote files and desktops using social VPN in a way similar to establishing a personal VPN.

Qiao *et al.* (2012) analyzed two possible C&C mechanisms for parasite P2P botnets and introduced their quasi-periodic characteristics. A detection framework was proposed with mathematical modeling based on quasi-periodic methods. Two algorithms were developed for this purpose, including a passive match algorithm (PMA) and an active search algorithm (ASA). ASA reduces time complexity significantly as compared to PMA. This approach was then implemented on eMule-like networks by evaluating some features of the packets which are used to send requests. This work is different from Gu *et al.* (2008a) in that it uses PULL mode to input parasites and communicate in the eMule-like networks. Communication in the P2P botnet is simple, as the botmaster sends commands to a single peer bot, which can be propagated to other peer bots in the P2P botnet. However, the management of P2P botnets is more difficult as compared to centralized C&C architecture. Furthermore, the slow response restricts the P2P botnet to be scalable. Similarly, it consumes less time than centralized architecture in sending commands (Raghava *et al.*, 2012). The elimination of the single point of failure and difficulty in detection are the implications of P2P botnets. Furthermore, detecting some of the peer bots does not reveal the failure of the whole P2P botnet. However, P2P botnets are slow in convergence and response, difficult to manage, and non-scalable.

Hybrid C&C architecture: The hybrid model inherits the properties of both centralized and decentralized/P2P architectures. The hybrid model is classified into two categories (Wang B *et al.*, 2010), servant bots and client bots. The servant bot acts as a client and a server simultaneously, which is configured with routable IP addresses (static IP); in contrast, the client bot does not listen to incoming connections as configured with non-routable IP addresses (dynamic IP). Servant bots send IP address information to the peer list and stay in listening mode to detect the port for incoming connections. Similarly, servant bots have additional responsibility to apply symmetric keys for each communication to stiffer the botnet detection mechanism.

2.4 Anomaly detection techniques vs. botnet detection techniques

The term 'anomaly detection' refers to the problem of finding exceptional communication patterns in the network traffic that do not conform to the expected normal behavior. These nonconforming patterns are often referred to as anomalies, outliers, exceptions, aberrations, surprises, peculiarities, or discordant observations in various application domains. Moreover, anomalies do not operate in a supervised environment. That is, in a broad sense, anomalies are unusual traffic patterns, explicitly or implicitly generated by various entities in an uncontrolled network environment. For example, a number of malware/anomaly detection systems have been presented in the literature (Bhuyan *et al.*, 2013; Tartakovsky *et al.*, 2013; Vaarandi, 2013).

Bhuyan *et al.* (2013) presented a comprehensive comparative survey of the literature on network anomaly detection. The evaluation criteria extracted for the deployment of network intrusion detection systems (NIDS) are primarily based on dataset assessment criteria and the detection strategy. For this purpose, the authors discussed various evaluation criteria for testing the performance and reliability of IDS.

Chandola *et al.* (2009) revealed that anomaly detection techniques can be classified according to different categorizations (classification based, nearest neighbor based, clustering based, statistics based, information theoretic approaches, spectral theory based, contextual anomalies, and collective anomalies). For each category of anomaly detection techniques, the authors made a unique assumption with respect to the notion of normal and anomalous data, and the effectiveness of the technique in a particular domain can be seen through applying a given technique to some specific domain.

In contrast, botnet detection refers to the detection of such malicious/anomalous activities that are governed in a controlled network environment. Malware distributors consider botnets a means to disseminate the malicious and anomalous activities around the globe. As a result, botnets became popular, constituting remotely controlled networks of hijacked computers. The basic aim of this distributed coordinated network is to initiate various malicious activities over the network, including phishing, click fraud, spam generation, copyright violations, key logging, and most importantly, DoS attacks. Botnets are identified as a serious threat to network resources over the Internet (Fossi *et al.*, 2011). Therefore, botnet detection is somewhat different from that of detection mechanisms posed by other malware/anomaly detection systems (e.g., IDS and IPS).

Anomaly detection techniques are out of the scope of this review. Therefore, we focus only on botnet detection techniques.

3 Review on the botnet detection phenomenon

This section presents thematic taxonomy of botnet detection techniques and reviews the latest detection techniques on the basis of anomaly based attributes of the taxonomy. Further, it investigates the advantages and critical aspects of botnet detection techniques.

3.1 Taxonomy of the botnet detection phenomenon

Researchers have developed many architectures and proposed different taxonomies to detect these malicious attacks (Feily *et al.*, 2009; Jing *et al.*, 2009; Zeidanloo *et al.*, 2010). Fig. 4 shows the taxonomy of botnet detection techniques which are classified based on their implementation.

Botnet detection techniques are classified into two broad categories (Liu *et al.*, 2008), IDSs and honeynets (Provos, 2004; Stinson and Mitchell, 2007). IDSs are further divided into anomaly-, signature-, and DNS-based IDSs (Stalmans and Irwin, 2011).

1. Honeynet: A honeynet is used to collect information from bots for further analysis to measure the technology used, botnet characteristics, and the intensity of the attack. Moreover, the information collected from bots is used to discover the C&C system, unknown susceptibilities, techniques and tools used by the attacker, and the motivation of the attacker. A honeynet is used to collect bot-binaries which penetrate the botnets (Freiling *et al.*, 2005; Abu Rajab *et al.*, 2006; Stinson and Mitchell, 2007). There are different techniques to capture bots in honeynets (McCarty, 2003; Freiling *et al.*, 2005; Abu Rajab

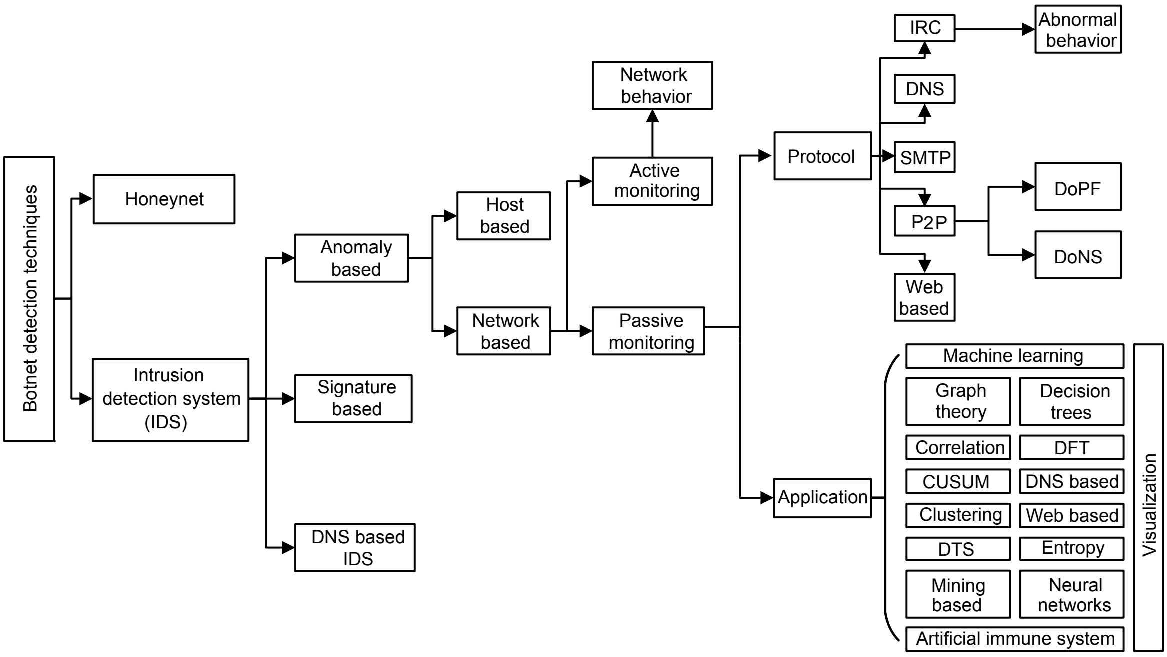

Fig. 4 Taxonomy of botnet detection techniques

et al., 2006; Dagon *et al.*, 2006; Ramachandran and Feamster, 2006; Barford and Yegneswaran, 2007; Oberheide *et al.*, 2007; Cremonini and Riccardi, 2009; Jing *et al.*, 2009; Kang *et al.*, 2009; Li *et al.*, 2009; Szymczyk, 2009; Rieck *et al.*, 2010; Pham and Dacier, 2011). However, intruders developed novel methods to overwhelm honeynet traps (Kugisaki *et al.*, 2007; Wurzinger *et al.*, 2009). Fig. 5 shows the honeynet architecture. The key component is honeywall, which is used to separate honeybots from the rest of the world. The honeywall is an L2/L3 device which acts as a gateway to pass through network traffic. The implications of a honeynet include simplicity in deployment, fewer resource requirements, minimal deployment cost, and usefulness in encrypted data. It can work under IPv6 environments.

However, the critical aspects of a honeynet include: (1) Scalability is limited, as it requires intensive hardware equipment (gateway routers and the honeynet system) be deployed; (2) Honeybots cannot anticipate finding Internet attacks and systems can track only malicious activities when interacting with it; (3) Discovery of the infected systems, placed as a trap is also challenging; (4) Sometimes attackers can take over honeybots to harm other systems or machines outside the honeynet (Bethencourt *et al.*, 2005; Zou and Cunningham, 2006).

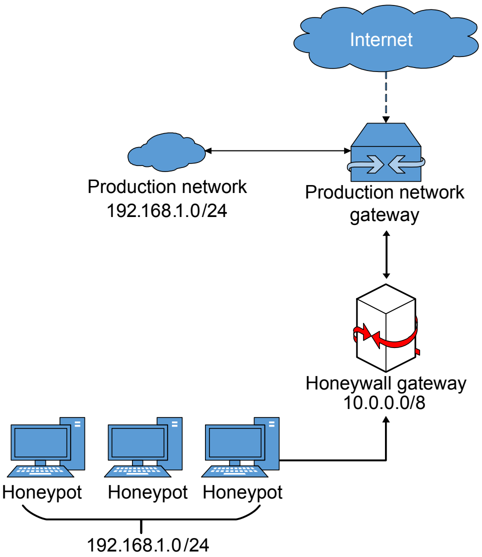

Fig. 5 Honeynet architecture

2. Intrusion detection system (IDS): An IDS is a software application or hardware machine to monitor system services for malicious activities or policy violations and report those violations to the management site. IDS detection techniques are further classified as two types of approaches, signature-based and anomaly-based. The advantage of an IDS

detection system (Goebel and Holz, 2007; Kugisaki *et al.*, 2007; Wurzinger *et al.*, 2009) is that it contains signatures of a number of known botnets. SNORT (www.snort.org) uses a signature-based detection scheme. However, limitations of IDS are as follows: IDS requires frequent updates of the knowledge base repository of signatures for detecting newly activated botnets (Kugisaki *et al.*, 2007). However, the refresh rate for IDS signature updates is small for the basic reason that anomalies are increasing rapidly; however, diagnoses are performed slowly. In contrast, signatures for such malicious attacks are not created at the same pace. Therefore, zero-day botnet attacks may not be detected, and it may not work for unknown botnets. Moreover, IDS detection techniques ignore identical bots with marginally different signatures.

Anomaly-based detection is a prominent research domain in botnet detection. The basic idea comes from analyzing several network traffic irregularities including traffic passing through unusual ports, high network latency, increased traffic volume, and system behavior indicating malicious activities in the network (Saha and Gairola, 2005; Binkley and Singh, 2006). Anomaly-based approaches are further divided into host- and network-based approaches. In host-based approaches (Stinson and Mitchell, 2007), individual machines are monitored to find suspicious actions. Monitoring is performed in terms of processing overhead, accessing kernel level routines, and changing system calls. Despite the importance of host-based monitoring, this approach is not scalable, as all machines are required to be fully equipped with effective monitoring tools, such as antivirus tools and spam detection software. In contrast, network-based approaches analyze network traffic in active or passive mode. In active monitoring, packets are injected into the network to measure the response time of the network (Wikipedia, 2013a), whereas in passive monitoring, network traffic is passed through specialized hardware devices to detect suspicious activity. Active monitoring is used to measure service quality by injecting test packets sent to the network, servers or applications. The additional traffic is generated by artificially injecting such packets which are not harmful for the performance of the system. The goal of active monitoring is to measure network parameters such as sampling techniques used, timing of packets, scheduling techniques used, packet type/size, monitoring of functions/path, and statistical quality. BotProbe (Wikipedia, 2013b) is an example of the active monitoring tool. A drawback of active monitoring is the increased network traffic payload based on the additional packets introduced into the network. In Strayer *et al.* (2008), an active monitoring tool examines network behavior based on different network characteristics, such as bandwidth, burst rate for botnet C&C evidence, and packet timing. It filters traffic that is unlikely to be part of botnet activity, classifies the remaining traffic into a group that is likely to be part of a botnet, and correlates traffic to observe common communication patterns that can lead to the detection of botnet activity. After examining 1.3 million real-time flows, the authors found the evidence of botnet activity to be compelling.

Stalmans and Irwin (2011) designed an IDS-based framework to detect the botnet based on malicious DNS queries and proposed a mitigation technique for malware infection on the network. Initially, the system was deployed at the core edge of the network to detect fast-flux domains, achieved by using a C5.0 decision tree classifier and a Bayesian statistical approach, with the assumption that positive labels are considered malicious domains and negative labels treated as legitimate traffic. The authors justified that their system could detect malicious domain names with a high degree of accuracy that would minimize the use of a blacklist. The passive monitoring approach uses specialized hardware to analyze network traffic. These devices are used specifically for anomaly detection. Similarly, existing network devices have the identical built-in functionalities (such as anomaly detection and signature-based IDS); for example, the latest routers and switches have the capability to monitor network traffic. Some specialized hardware devices dedicated for monitoring purpose are also available. For example, NetScout (www.netscout.com) and Panda Firewall (Panda Security, 2013) are the two prominent hardware systems which use passive monitoring for the detection of malicious network traffic. Passive monitoring does not increase the network payload while monitoring network traffic, because it does not inject extra packets into the network. However, the drawbacks of passive monitoring include the following: polling is required to collect data for the purpose of monitoring traffic, which substantially increases the network payload. Therefore, it seems to be enormous if a device captures each packet for flow analysis and the passive technique is required to view all traffic; thus, it also involves security and privacy issues.

3.2 Review on botnet detection techniques

This section reviews the latest botnet detection techniques through using thematic taxonomy.

3.2.1 Host-based botnet detection techniques

Gu *et al.* (2009) proposed an anomaly detection IDS, which produces a low false positive rate. A server-based approach is employed for anomaly detection while reducing false positive alarms in the network. Two approaches are combined for anomaly detection, including host-level anomaly detection and proliferation of the false positive rate. The Markov model is employed for anomaly detection. This suggested approach correlates malicious instances at the destination, which is considered a major drawback.

The behavior of bots is investigated by scanning the processes relevant to specific applications installed on the host machine (Stinson and Mitchell, 2007). Each bot independently initializes commands received from the C&C system, whereas each command includes certain parameters, specific types, and predetermined execution orders. BotSwat (Stinson and Mitchell, 2007) is a tool for monitoring home operating systems (such as Windows XP, Windows 2000, and Windows 7) and recognizing the home machines anticipated as bots. Initially, BotSwat acts as a scanner, monitoring the execution status of the Win32 library and observing runtime system calls created by a processor. Furthermore, it tries to discover bots with generic properties despite the particular C&C architecture, communication protocols, or botnet structure. The problem with this approach is the lack of security for system calls.

Masud *et al.* (2008) developed an effective host-based botnet detection technique using a flow-based detection method by correlating multiple log files installed on the host machines. As bots normally respond more quickly than humans, mining and correlating multiple log files can be easily realized. It is proposed that these techniques can be efficiently performed for both IRC and non-IRC bots, by correlating several host-based log files for some C&C traffic detection. Liu *et al.* (2008) proposed a bot which has a life cycle in three different phases (startup, preparation, attack). In the startup phase, the bot is automatically initiated without requiring any user input. The initialized bot establishes a connection with its botmaster through a C&C channel in the preparation phase. Afterward, the bot eventually initiates both local and remote attacks. BotTracer (Liu *et al.*, 2008) was developed to detect these phases with the help of virtual machines. BotTracer attempts to discover the channels through which the bot actively establishes a connection with the C&C network. After capturing those channels, it compares them with the known properties of the C&C channels on which the botnet traffic is moving. A basic drawback of BotTracer is that, it cannot detect the existence of virtual machines. BotTracer also continuously monitors vulnerabilities in the system calls for any potential botnet activity.

The multi-agent bot detection system (MABDS) (Szymczyk, 2009) is a hybrid technique which associates an event-log analyzer with the host-based intrusion detection system (HIDS). This uses multiagent technology which combines the administrative agent, user agent, honeypot agent, analysis of the system, and the knowledge database. The basic problem for this technique is the slow convergence of new signatures with the knowledge base. HIDS (Ying *et al.*, 2010) consists of log analyzer technology along with a back-propagation (BP) neural network. Based on misuse detection, the host-based technology is introduced, and the BP neural network is an approach for anomaly detection. It is shown that the intrusion detection system outperforms existing detection systems by combining these two technologies. Scalability is the major concern of this technique. DeWare (Xu *et al.*, 2011) is a host-based security tool which provides a host-based security mechanism by enforcing inference rules to check the correct dependency properties of the calls of an operating system (OS) or the file system being accessed. The ultimate security concern for this technique is that user-level OS routines may be intercepted with kernel-level routines, which may cause the OS to malfunction.

The hybrid intelligent intrusion detection system (HIIDS) (Murugan and Kuppusamy, 2011) implements the neural network approach to mine the malicious attack definitions. It provides the attack detection mechanism by applying data mining techniques on botnet behavior. After collecting information on the behavior of certain attacks, this information is passed through a decision support system based on fuzzy inference rules. Moreover, the combination of fuzzy inference rules with the neural network provides an efficient and accurate intrusion detection technique. This approach, however, has a limitation: it does not provide autonomous learning of inference rules in its decision support system. Ge *et al.* (2012) proposed a host-based intrusion detection system to detect cyber-attacks in mobile ad-hoc networks (MANETs). To achieve effective detection and minimal impact on the network, a random sampling based technique and stratified sampling technique were proposed to uniformly sample the information detected. It is shown that the stratified sampling technique produces better results than a simple random sample technique. This approach can work only with mobile ad-hoc networks.

ELM (Creech and Hu, 2013) is another approach proposed based on a contiguous/discontiguous system call design. ELM was proposed to reduce the false alarm rate while increasing the anomaly detection rate. In addition to that, it can observe the kernel-level routines in the OS and highlight those activities generated by high-level languages, to better understand the anomaly behavior of a program. Furthermore, this technique is resilient to anomaly attacks and offers portability among different OSs. A new Linux-based dataset called 'ADFA-LD' is now publically available.

A host-based technique is used to check whether the individual machine is infected by the bot or not. Each bot affects the individual machine by changing its registry structure, system calls, and system files. One advantage of using a host-based technique is that it can easily avoid download attacks and especially for those attacks attempting at start-up (Xu *et al.*, 2011). The protection at the host level can be provided by scanning individual machines in an organization; however, this is considered to be a time consuming and costly task.

Host-based botnet detection approaches are summarized in Table 2.

3.2.2 Network-based botnet detection techniques

In a network-based botnet detection strategy, the malicious traffic is perceived by observing the network traffic within different parameters, including network traffic behavior, traffic patterns, response time, network load, and link characteristics. Network-based approaches are further classified into two types, active monitoring and passive monitoring.

Active monitoring: In active monitoring botnet detection policy, new packets are injected to detect malicious activities in the network. Usually this technique is not considered a preferable strategy due to additional load to network traffic. BotProb (Tokhtabayev and Skormin, 2007) is considered an active monitoring strategy, which injects packets into the network payload for finding suspicious activity caused by humans or bots. As non-human bots usually transmit commands on a predetermined pattern, which corresponds to the cause and effect correlation between C&C and the bots. Such a command and response architecture can easily determine the existence of bots because the response comes from the predetermined command behavior.

The basic property of response time distinguishes active monitoring from passive monitoring and differentiates active monitoring in terms of slow detection response, multiple infection detection stages (such as BotHunter (Gu *et al.*, 2007)), several rounds of communication activities (such as BotSniffer (Gu *et al.*, 2008a)), and a longer communication response time (such as BotMiner (Gu *et al.*, 2008b)). BotProb is not intended to overcome the existing passive detection techniques; instead, this technique works from a different perspective. The critical aspect of the active monitoring technique is that it overloads the usual network traffic due to the additional packets injected into the network. Moreover, it is difficult to separate legitimate traffic from the artificially injected traffic for anomaly detection, which disrupts the routine traffic and is subjected to the privacy issues. In a bot life cycle, each botmaster actively tries to connect to their bots and initiate some commands; this process seems to be common for a majority of botnet attacks. Moreover, the same network protocols are used to perform malicious activities (Trend Micro, 2006).

Passive monitoring: In passive monitoring, network traffic is observed when the data is passed through the medium. The network traffic is analyzed by applying different anomaly detection techniques, including distance based techniques, support vector

Table 2 Host-based botnet detection techniques: a timeline

Proposed model	Methodology	Shortcoming
Non-stationary Markov models (Tokhtabayev and Skormin, 2007)	Host-based detection by monitoring system calls	Correlates worm instances at the destination
Remote control behavior of bots (Stinson and Mitchell, 2007)	Content-based and substring-based tainting	Rarely exhibits the external control behavior
BotSwat (Stinson and Mitchell, 2008)	Monitors the Win32 library	Does not consider the C&C communication protocol or specific botnet structure
Mining multiple log files (Masud *et al.*, 2008)	Flow-based detection through data mining	Privacy and security issues
BotTracer (Liu *et al.*, 2008)	Three-phase model/flow-based	Unable to detect virtual machines
Multi-agent bot detection system (MABDS) (Szymczyk, 2009)	Hybrid model (host IDS + OS event log analyzer)	New signature update problem
HIDS (Ying *et al.*, 2010)	Hybrid model (log file analyzer + BP neural network)	Non-scalable
DeWare (Xu *et al.*, 2011)	Enforces the rules on OS routines	Kernel-level OS routines could be intercepted
Hybrid intelligent intrusion detection system (HIIDS) (Murugan and Kuppusamy, 2011)	Application of fuzzy logic through network profiling, hybrid model	Autonomous learning of the inference rules
Stratified/Random sampling techniques (Ge *et al.*, 2012)	Steadiness, the tradeoff between detection accuracy and bandwidth overhead incurred	Specifically used for MANET
A semantic approach to host-based intrusion detection (Creech and Hu, 2013)	Based on contiguous/discontiguous system call patterns	Needs to investigate the transference process

machines, neural networks, cluster analysis, and learned association rules (Wikipedia, 2013a). It does not require any additional traffic be injected into the network as in active monitoring. This approach passively monitors network traffic with the idea that, botnet traffic has similar characteristics which can be easily detected by observing the request-response behavior of traffic during some specified time span. The basic theme of passive monitoring is that, traffic tends to respond in the same communication pattern in the botnet, despite the architecture being employed (client server or peer-to-peer). A pre-configured program is installed on each bot in the botnet, which responds in a similar fashion. Passive monitoring techniques can be further classified according to their applications and protocols (Silva *et al.*, 2013). Protocol-specific passive monitoring techniques include P2P, HTTP, SMTP, and DNS.

A number of botnet detection techniques have been proposed based on passive monitoring using various application models (Dagon *et al.*, 2007; Gu *et al.*, 2007; 2008b; Iliofotou *et al.*, 2007; Mukosaka and Koike, 2007; van Ruitenbeek and Sanders, 2008; Barsamian, 2009; Chang and Daniels, 2009; Ha *et al.*, 2009; Kaemarungsi *et al.*, 2009; Kang and Zhang, 2009; Kang *et al.*, 2009; Leonard *et al.*, 2009; Lu *et al.*, 2009a; Mansmann *et al.*, 2009; Perdisci *et al.*, 2009; Shahrestani *et al.*, 2009; Wang CD *et al.*, 2009; Wang W *et al.*, 2009; Coskun *et al.*, 2010; Huang *et al.*, 2010; Jiang *et al.*, 2010; Liao and Chang, 2010; Liu *et al.*, 2010; Yu F *et al.*, 2010; Chen *et al.*, 2011; François *et al.*, 2011; Stringhini *et al.*, 2011; Thonnard and Dacier, 2011; Zeng *et al.*, 2011; Zhang *et al.*, 2011b; Sanchez *et al.*, 2012). The passive monitoring techniques employing various application models include statistical approaches, graph theory, machine learning, correlation, entropy, stochastic model, decision trees, discrete time series, Fourier transformation, group based analysis, data mining, clustering approach, neural networks, visualization, and a combination of these technologies.

What makes a botnet more of a threat than others? To help compare botnets, we identify key metrics for measuring the utility of a botnet, based on their use. Using these performance metrics, we consider the ability of different response techniques to degrade or disrupt botnets. In the following we will discuss the above listed application-specific passive botnet detection techniques and analyze their rationale and shortcomings with respect to their implementation metrics along with their future directions.

From Table 3 to Table 18, we have critically analyzed each botnet detection application model according to their rationale, weaknesses associated with each technique, specifically, focusing on parameters/metrics (if any), and future directions to improve respective techniques. Moreover, we try to capture more recent research exposure to make readers aware of the most up-to-date trends towards botnet detection. The basic motive behind this critical study is two-sided: it is easy to grasp the basic knowledge about the published research schemes, but difficult to state and discuss each technique separately in a single article.

1. Statistical botnet detection techniques: Botnet detection techniques based on statistical modeling have been discussed in the literature (Barsamian, 2009; Kaemarungsi *et al.*, 2009; Wang CD *et al.*, 2009; Liu *et al.*, 2010; Zhang *et al.*, 2011b; Silva *et al.*, 2013). Barsamian (2009) proposed a framework for characterizing network behavior on an Ethernet network. The approach assures conformity to the existing signatures and detects reliable changes in the behavior of the botnet. Moreover, they provided reliable methods for detecting periodic and synchronous behavior based on a *K*-means approximation. Botnet detection based on statistical analysis of mail flow (Wang CD *et al.*, 2009) enhances the speed of email filtering while reducing network traffic and potentially minimizing the false positive rate. The shortcoming of this approach is that it does not filter out the content of the email but the email header. Additionally, Kaemarungsi *et al.* (2009) developed a tool for analyzing botnet statistics using the data shared by the shadow-server foundation. A statistically comprehended view on botnet statistics is presented; however, the lack of a complete picture (partial view of the botnet) on the Internet is the critical aspect of this approach, which is missing. Liu *et al.* (2010) revealed a P2P botnet detection mechanism based on network stream analysis. Their technique is based on the following three algorithms: (1) P2P node detection algorithm, (2) P2P node clustering algorithm, and (3) similarity detection algorithm. The lengthy process to identify botnets through network stream analysis discourages this technique to be implemented in real-time environments.

Rrushi *et al.* (2011) proposed a virulence estimation approach based on random sampling along with a novel statistical learning technique and gave a botnet-versus-network setting. Mathematical modeling using Matlab was employed to conduct experiments and validate results using GTNetS (a realistic simulator). The infection rate and network vulnerability are the key factors in this research. Unlike Choi *et al.* (2010), Rrushi *et al.* (2011) claimed that maximum likelihood estimation was used in their statistical approach to botnet detection. Marko and Vilhan (2012) proposed a novel technique which contributes to the ability of monitoring botnet's nodes on a local area network (LAN) through observing DNS queries. After deployment of the proposed rules, eight suspicious bots were identified by looking at their DNS records (log) for the nodes. Among them three were SPAMMERS, identified due to the large number of DNS mail exchange (MX) queries. The five nodes were depicted as malicious nodes as these were involved in querying pseudo-random generated domain names. It was predicted that this technique can restrict a large number of bots during the process of obtaining instructions from their C&C in a reasonable amount of time.

Sousa *et al.* (2012) installed spam botnets to capture network traffic and characterize this network traffic in order to identify the main activities. After intensive statistical analysis, variations were observed in the behavior/features (temporal evaluation, protocol variations) of different spamming botnets, which can further be explored to design various spam botnet detection techniques. Three spam botnets (Grum, Cutwail, and Bobax) were installed and evaluated after intensive statistical analysis. It was concluded that all botnets contain some distinct features that can be explored to develop different botnet detection techniques. The relevant metrics for this evaluation are the number of packets per hour, download/upload protocols, and the number of unique peers per hour. Table 3 presents a summary of statistical modeling based botnet detection techniques and shows their implications and critical aspects.

2. Visualization based botnet detection techniques: The botnet detection techniques based on a visualization approach have been highlighted in the literature (Mukosaka and Koike, 2007; Kang *et al.*,

2009; Mansmann *et al.*, 2009; Shahrestani *et al.*, 2009). Dorothy (Cremonini and Riccardi, 2009) is an open framework that allows for observing the botnet activity after characterization of the botnet behavior by applying a set of parameters (size, structure, life cycle). This approach embeds the infiltrate module (a tool used to encapsulate the features of an IRC client allowing for joining in an IRC channel) and a data visualization module (graphical representation of botnet behavior, which is the key contribution of the article). Mansmann *et al.* (2009) presented TreeMap and graph representation through intensive network flow analysis. The usability of this technique was described through three case studies, including analysis of service usage in a network, detection of a distributed attack, and identification of hosts that are susceptible to communication with external IPs (for malicious activity). The proposed architecture is not viable for large/real-time datasets because of the complex architecture that must be adopted in the approach.

Shahrestani *et al.* (2009) presented a combination of data mining and visualization for network flow analysis, wherein the malicious data was passed through several trust models, and after re-evaluation of the flows, the data was aggregated to detect malicious traffic through visualization. For large-scale LANs, Mukosaka and Koike (2007) presented the visualization technique for security mechanism. A strong filtering mechanism was provided along with 3D visualization. This proposed system integrates logical, geographical, and temporal information in a single 3D visualization manner. Moreover, the system provides a strong filtering mechanism. A larger false detection rate is a critical aspect of this model. It lacks the support for inbound traffic and relies on minimum

Table 3 Analysis of statistical botnet detection approaches

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
A framework for characterizing network behavior on an Ethernet-protocol network (Barsamian, 2009)	Conformity to the signature and detect changes in behavior. Detecting periodic and synchronous behavior based on *K*-means approximation	Contention that botnets can evade detection by moving away from the IRC-based C&C architecture	-	Knowledge base, automatic correlation of flow, behavioral fingerprinting
Botnet detection based on analysis of mail flow (Wang CD *et al.*, 2009)	Enhancing the speed of e-mail filtering reducing network delay, low false positive rate	- Filter rule is not based on contents of email - Large transmission delay of regular emails	- Judged spams - Judged legitimate emails	Reduction in transmission delay of regular mails, parameter randomness
Development of a tool for analyzing botnet statistics using the data shared by the shadow-server foundation (Kaemarungsi *et al.*, 2009)	- Handling the botnet threat using shadow server information - Presenting a statistical view	Partial view of the botnets over the Internet	-	Collaboration among CERTs
P2P botnet detection based on network streams analysis (Liu *et al.*, 2010)	Detection model comprises: - P2P node detection algorithm - P2P node clustering algorithm - Similarity detection algorithm	- Results taken from the LAN simulation environment - Lengthy process to locate botnet from network stream analysis	- Polymorphic - Undiscovered - Cryptographic channel	This algorithm should be tested for Sinit, Phatbot P2P botnets
Detection of P2P botnets using statistical traffic patterns (Zhang *et al.*, 2011b)	- Estimating active time based on flow-clustering - P2P fingerprints - Detection algorithm	- Randomness in P2P communication patters - By exploiting the P2P threshold	Statistical fingerprints	Making evasion harder by combining different botnet detection techniques
Generic Feature Selection (GeFS) measure for botnet malware detection (Silva *et al.*, 2012)	- A comparison study between GeF-SCFS and genetic-algorithm-CFS and with best-first-CFS methods - 99.9% irrelevant and redundant features could be eliminated from the dataset	Full-set and GACFS approaches are not good in the sense that they produce a very high false positive rate and contain a large feature set, which is difficult to analyze	- Statistical properties of the dataset - Linear vs. non-linear correlation - Static and dynamic approaches	-

information for anomaly detection. Table 4 summarizes the visualization techniques for botnet detection and presents their implications and critical aspects.

3. Data mining based botnet detection approaches: BotMiner (Gu *et al.* 2008b) is an extension of BotSniffer (Gu *et al.*, 2008a), which is used to detect real-world botnets including IRC-based, P2P-based, Nugache, and Storm worm with a low false positive rate. Masud *et al.* (2008) tested a network flow based framework by mining multiple log files to detect bot activities in user machines by temporally correlating two user log files (tcpdump, exedump). These log files were used to record all incoming and outgoing network traffic (packets) and also to maintain the history of the start time of each application execution at the user level of machine. This approach needs to be implemented on real-time systems to validate the effectiveness of the framework.

Another P2P botnet detection methodology presented in Liao and Chang (2010) implements P2P botnet detection based on a data mining technique to analyze network behavior at the gateway level. A significant aspect of this approach is that it can monitor encrypted network traffic. However, this scheme is made for small-scale network infrastructures (LANs), and thus cannot be deployed or validated on large-scale networks. Similarly, it uses network address translation (NAT) mechanism to route network traffic; however, NAT is not efficient in detecting P2P network flows (Jelasity *et al.*, 2011); moreover, it cannot scan traffic contents. Table 5 summarizes traffic mining based botnet detection approaches and presents their implications and critical aspects.

4. Graph theory based botnet detection techniques: Graph theory provides mathematical structures to model pair-wise relationships between different entities (objects); therefore, it is a significant technique for botnet detection. An executive summary is presented in Table 6.

Dagon *et al.* (2007) used the taxonomy of botnet structures to identify key metrics from different response techniques (Erdos-Renyi random graph model, Watts-Strogatz small-world model, and Barabasi-Albert scale free model) for identification of various malicious activities (spam, DDoS, etc.). Real-time and simulation results showed that random network models provide considerable stability to botnets. Iliofotou *et al.* (2007) introduced traffic dispersion graphs (TDGs) as a means to monitor, analyze, and visualize network traffic in response to malicious activities. TDGs model the social behavior of communication (“who communicates to whom”), whereas edges represent interactions (such as exchange of a certain type or the number of messages/packets) between parties. The critical aspect is that it employs restricted access layer support, and thus the model is not deployable in public organizations. Similarly, port-based TGDs are used to identify the type of application on a given port; thus, massive hits on a specific port trigger an anomalous behavior of requests. Therefore, it is difficult to define a threshold for the applications.

Table 4 Analysis of visualization techniques for botnet detection

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
The Dorothy Project (Cremonini and Riccardi, 2009)	Infiltrate module & data visualization module	-	Botnet size, hidden structure, botnet life cycle	-
TreeMap and graph representation technique (Mansmann *et al.*, 2009)	NetFlow analysis using TreeMap and graph representation	Lack of real-time data collection	Analysis of service usage, distributed attack detection, malicious host investigation	Real-time data collection & analysis
Network flow analysis (Shahrestani *et al.*, 2009)	Combination of two approaches (data mining & visualization)	No analysis of traffic contents, covering theoretical aspects only	Traffic flow characteristics (static, dynamic)	Implementation of this model may prove its validity
Visualization system for security mechanism in large-scale LAN (Mukosaka and Koike, 2007)	3D visualization, provision of strong filtering mechanism	False detections, not for inbound traffic, limited information for detection	IP matrix, logical, temporal & geographical information	Adding more filters to avoid false detections, periodically updated results

Table 5 Analysis of traffic mining based botnet detection approaches

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
BotMiner (Gu *et al.*, 2008b)	Extension of BotSniffer C-Plane: detection of C&C A-Plane: detection of malicious activities	It is not designed for special kinds of protocols	Working under both centralized and decentralized environments	SMTP and HTTP support
Flow-based technique for mining multiple log files (Masud *et al.*, 2008)	Temporal correlation between log files (tcpdump, exedump)	It does not trace IRC flows	Vector machines, decision trees, Naïve Byes, boosted decision trees	Implementation on system level logs, real-time experiment
P2P botnet detection using a data mining scheme (Liao and Chang, 2010)	A P2P botnet detection method relying on monitoring traffic at the gateway and using data mining technology to analyze network behavior	- It works only within a LAN environment; it should be distributed to the ISP level to detect P2P botnets in a large-scale network - Existence of NAT technology makes it difficult to detect P2P flows	- Encrypted packets - Pre-warning mechanism - Sophisticated detection of botnet flow	Large-scale network design for better botnet detection

Table 6 Analysis of graph theory based botnet detection approaches

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
Taxonomy of botnet structure (Dagon *et al.*, 2007)	Responsive technique, random models	Not a detection technique	Giant portion, average available bandwidth, diameter, local transitivity	Refined metrics, higher accuracy of results required
Traffic dispersion graphs (TDGs) (Iliofotou *et al.*, 2007)	Measurement of social interaction of network hosts	Restricted access layer support, edge filters, designing thresholds for applications	Aggregation, ability to spot patterns	Deploying it at the access link, applying it to different edge filters, live deployment of TDGs
BotTrack (François *et al.*, 2011)	Analysis of communication behavioral patterns	Non-scalable	Clustering technique	Inferring potential botnet activities
DNS failure graph analysis approach (Jiang *et al.*, 2010)	Lightweight anomaly detection technique	Based on unpredicted DSN traces	Tri-nonnegative factorization technique	
Stealthy botnet characterization (Leonard *et al.*, 2009)	Graph based model for detecting botnet C&C mechanism	Simulation study, attack defense interaction	Botnet detection ratio and resilience	Requiring model validation in the real-world testbed
P2P structural detection (Ha *et al.*, 2009)	Detection and mitigation testbed	Focusing mainly on P2P	Kademlia algorithm	Focusing on other botnet architectures
IRC botnet detection (Wang W *et al.*, 2009)	Based on channel distance, not requiring binary analysis, reduced delay, 4-tuple model	Difficulty in detecting nicknames with variable length random numbers	IRC nicknames, detect-point value, detection threshold	Considering more structural factors
Infrastructure level detection measurement (Zeng *et al.*, 2011)	Exploiting structural properties of botnet from graph analysis in high level infrastructure	Misclassification of traffic, real-dataset constraints	AS, PoP (point of presence), router rendezvous	Resolving detection complication for modified chord
P2P botnet detection using mutual contacts (Coskun *et al.*, 2010)	Random peer selection model	Cannot work with structured P2P topology, poisoning clusters	Mutual contacts	Considering other P2P botnet architectures

Jiang *et al.* (2010) proposed a lightweight malicious traffic detection approach to identify suspicious activities through a DNS based failure graph. A graph decomposition algorithm was employed based on a tri-nonnegative matrix factorization approach which gradually extracts coherent sub-graphs from the failure of DNS queries. As this approach relies on unpredicted DNS traces, it is not necessary that every unpredicted DNS request should lead to some malicious activities (Cisco Systems, 2012). A study based on stealthiness of botnets was provided by Leonard *et al.* (2009), looking into the survival of botnets. The study focused on finding the properties of stealthy botnets which survive against detection mechanism. For this purpose, a graph based model is deployed in a simulated environment to diagnose stealthiness and the location of C&C. The critical aspect of this approach is to measure the stealthiness of botnets in response to the attack. It cannot anticipate the attack model or detect zero-day level botnet attacks. Apart from a simulation study, experiments were also performed on real-work scenarios to validate the effectiveness.

Ha *et al.* (2009) carried out an extensive simulation study to detect P2P botnets, running in Kedmelia (a protocol for P2P networks). The reachability, scaling, and clustering properties of P2P botnets were analyzed using graph theory. Monitoring botnet activity in this way was determined to be a difficult task. Moreover, the usefulness of some well-known attacks in response to P2P networks was evaluated, which resulted in various mitigation approaches such as content poisoning, Sybail based mitigation, and eclipse based mitigation techniques. Wang W *et al.* (2009) discussed other novel approaches to detecting IRC botnets. One positive aspect of the approach is the detection of zero-day botnets without any delay and it does not require any pre-scanning of existing bots. Channel distance is considered a primary metric corresponding to any IRC channel nickname (which is composed of random numbers and different letters). A new algorithm was proposed to efficiently detect botnets based on the channel distance. Furthermore, experiments were carried out to validate the performance of this algorithm. According to the experiment results, the derived detect-point value was 20 and the detection threshold value was 0.5. Zeng *et al.* (2011) proposed large-scale P2P botnet detection based on graph theory. This study focused on identifying a trade-off between stealthiness and the resilience of botnets. Real-world datasets were used; therefore, the accuracy of such datasets depends upon the characterization of the behavior of the dataset. Similarly, results may be misleading due to outdated datasets, as they may not represent the current state of the Internet because of the rapid growth of the Internet.

In the random peer selection model (Coskun *et al.*, 2010) there is a simple scheme for identifying potential candidates of unstructured P2P botnets by identifying a few peers in a network. The authors formulated the problem in graph theory and used an iterative algorithm. However, it is easy to evade the approach by deploying structured P2P network topology because it works only for unstructured P2P topologies.

5. Botnet detection techniques based on clustering: Perdisci *et al.* (2009) proposed a passive anomaly detection framework to track and detect malicious fast-flux service networks based on recursive DNS traces collected from multiple networks. The distinctive aspect of this approach is that it can detect malicious flux service networks spontaneously, which can be accessed by those users who fall victim to suspicious contents promoted by instant messaging spam, blog spam, social website spam, and cloud spam, instead of regular email spam. Since this approach is adopted for passively monitored real-time analysis and keeps noticing the activities of real users, it slows down the network.

Lu *et al.* (2009a) proposed a new hierarchical framework which can automatically discover malicious botnets in large-scale networks. The network traffic was classified into different application communities based on payload signatures and cross-association clustering algorithms. Temporal frequent properties of traffic flows were analyzed after recognizing various application communities. The resultant flow was differentiated as malicious channels created by bots from the normal network traffic generated by Internet users. This approach is effective for IRC traffic and therefore web communities can take advantage of this scheme. Another scheme (Chang and Daniels, 2009) detects C&C channels of P2P botnets based on behavior profiles (temporal and spatial correlations) of the nodes. Using node behavior clustering it captures the normal traffic traces.

Two anomaly detection techniques were proposed based on statistical analysis of prominent behavior clusters. Evaluation was performed on simulated and realistic environments and validation performed on real datasets collected from the enterprise network. The deficiency of this approach is that it measures behavior clusters at different times, as the behavior cluster behaves differently for different timeframes and workloads. For instance, backup of data is usually done during the nighttime.

SBotMiner (Yu F *et al.*, 2010) was proposed to detect botnet traffic from search engine query log files on a wide scale. This system identifies bot generated search traffic by observing query logs. SBotMiner was proposed to detect stealthy bot traffic which is normally difficult to identify by off-the-shelf software/tools. Additionally, individual queries may be generated in an indistinguishable form that is most relevant to the normal query pattern; therefore, the pattern of such queries often shows similar or common context if viewed in the aggregate. The problem with this approach is that it cannot detect diverse search requests/ambiguous queries (Welch *et al.*, 2011). The botnet detection approaches based on clustering approaches are summarized in Table 7.

6. Correlation based botnet detection approaches: Thonnard and Dacier (2011) presented a strategic analysis of spam botnet operations based on finding the association between botnets through their spam campaigns. This study is focused on identifying similarities or differences in their modus operandi. It provides an in-depth analysis of the strategic behavioral characteristics of spamming botnets by observing their aggregate spam campaigns. The usefulness of evolving attack attribution methodologies was revealed to extract astuteness from large spam datasets and to associate spam campaigns according to various combinations of different features. It was found that different botnets have some tight relationships between different botnet families (for instance, Grum/Rustock or Maazben/Lethic). Moreover, differences were found in spam campaigns performed by other bots such as Lethic versus Rustock, Xarvester, or Bagle. A more robust correlation architecture was proposed by Zhang *et al.* (2011a) to perform botnet detection for high speed and high volume networks, including a botnet-aware adaptive packet sampling algorithm along with a scalable spatial-temporal flow correlation mechanism. However, evasion is easy for the botmaster once it recognizes the proposed algorithm.

Table 8 summarizes the botnet detection approaches based on correlation.

7. Stochastic function based botnet detection approaches: A stochastic model for creation of P2P botnets was presented in van Ruitenbeek and Sanders (2008). The model was motivated from the Storm worm botnet and considered the most general model of P2P botnets. Simulation results demonstrated the effectiveness for both prevention measures (user education and antivirus products) and disinfection and detection methods (removal products or rootkit detection).

Table 9 shows the botnet detection techniques based on the stochastic function.

Table 7 Analysis of clustering based botnet detection approaches

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
Recursive DNS traces (RDNS) (Perdisci *et al.*, 2009)	Passive technique to detect malicious flux network	Performance degradation	Fast flux domains	Secure edge router from unauthorized access to internal resources
Automated detection on large-scale network (Lu *et al.*, 2009a)	A hierarchical framework to automatically discover botnets	Working only for IRC and the web community	Payload signatures, cross-associative clustering algorithm, temporal-frequent characteristics of flows	Considering P2P, SMTP, and the email community
Behavior clustering (Chang and Daniels, 2009)	Characterization of node behavior by jointly considering spatial and temporal correlations	Behavior profiling and detection at different times	False positive rate	Traffic characterization, time variant clustering, more traffic traces
SBotMiner (Yu F *et al.*, 2010)	Botnet detection from search engine query logs	Unable to detect diverse search requests	Log files, IP addresses, queries	Working for abnormal search traffic

8. Entropy based botnet detection approaches: The spatial snapshot fast flux detection system (SSFD) (Huang *et al.*, 2010) is a real-time fast flux service network (FFSN) botnet detection system. It reveals FFSNs through acquiring the geographic traffic patterns of network hosts and mapping the IP address of a DNS response in a geographic coordinate system. The system uses spatial distribution estimation to assess the uniform geographic distribution of infected hosts. The spatial service relationship improves the misclassification between content distribution systems/networks and FFSN. The critical aspect of SSFD is that it cannot work for dynamic DNS. The multi-chart CUSUM algorithm (Kang and Zhang, 2009) is a new entropy based method for detecting Storm botnets. Small-scale experiments (fault-positive and fault-negative) have shown that this approach can effectively detect Storm botnets. A wide-scale study is needed to validate the results for wide-area networks.

Table 10 shows the summary of entropy based botnet detection techniques.

9. Decision tree based botnet detection approaches: BotCop (Lu *et al.*, 2009b) is an online botnet traffic classification approach, in which network traffic is classified according to various application communities (char, P2P, web). Similarly, the frequent temporal characteristics of network flow are analyzed and studied to separate the malicious network communication from the usual network traffic generated by normal users. This approach is used specifically for IRC-based communication. An IDS-based framework (Stalmans and Irwin, 2011) was designed to detect botnets based on malicious DNS queries, and a mitigation technique was proposed to address malware infection on the network. Initially, the system is deployed at the core edge of the network to detect fast-flux domains using a C5.0 decision tree classifier and a Bayesian statistical approach, with the assumption that positive labels are considered malicious domains and negative labels treated as legitimate traffic. The authors justified that their system can detect malicious domain names with high accuracy, which minimizes the need to use a blacklist.

Table 8 Analysis of correlation based botnet detection approaches

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
A strategic analysis of spam botnets (Thonnard and Dacier, 2011)	Triage methodology	Misperception of results	Bot signature, OS details, source IP, URI domains	Classifying unknown bots
Incremental LS-SVM learning (Chen *et al.*, 2011)	Detection of encrypted botnet communication	Focusing on online-learning algorithms	Server IP addresses	Diverse application experimentation
Botnet detection using adaptive traffic sampling (Zhang *et al.*, 2011a)	Two-fold approach: B-sampling and cross-epoch correlation	Evasion is possible once the algorithm is captured	Packet payload information	Making evasion harder by combining different complementary detection techniques

Table 9 Analysis of stochastic botnet detection approaches

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
BOTMAGNIFIER (Stringhini *et al.*, 2011)	Detection of spamming bots	Small dataset deployed	Seed pools, transaction log	New data input, more comprehensive transaction log
Modeling P2P botnets (van Ruitenbeek and Sanders, 2008)	Detection of Storm worm attacks	Increased rate of infection than detection	Propagation of active or inactive bots	Effectiveness of other potential anti-malware techniques

Table 10 Analysis of entropy based botnet detection approaches

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
Fast-flux detection based on snapshot mechanism (Huang *et al.*, 2010)	Spatial snapshot fast-flux detection system (SSFD)	Cannot work for dynamic DNS	Spatial distribution estimation, spatial service relationship evaluation	Zero-day FFSN problem, sleep domain problem
Application entropy theory using multi-chart CUSUM (Kang and Zhang, 2009)	Information entropy for multi-chart CUSUM, Kaulfman algorithm	Lack of real-world experimentation	Net flow characteristics (TCP, UDP, ICMP)	Working in large-scale environments

Table 11 shows the summary of decision tree based approaches.

10. Botnet detection approaches based on machine learning: Chen *et al.* (2011) proposed an algorithm based on an incremental least-squares support vector machine (LS-SVM) learning scheme, and evaluated the performance on two real-world datasets. It focuses on the detection issue by using an online learning scheme that can be used for both training sets and evolving features. Finding malicious bots depends on how often a client machine visits a server machine by looking at the IP addresses of the server machines. Moreover, this approach can detect encrypted botnet communication. However, this scheme targets only online-learning systems/algorithms.

Sanchez *et al.* (2012) proposed a support vector machine (SVM) based classification approach to separate end user (EU) malicious spam from the legitimate mail server (LMS) based on a set of machine features that cannot be easily evaded by spam initiators. It was concluded that their approach achieves detection accuracy up to 99.27%, with a minimum false positive rate of 0.44% and a false negative rate of 1.1%. The results were validated using eight different DNS-based blacklist models. Small-scale dataset experimentation restricts this approach from being deployed on large-scale network infrastructures. Livadas *et al.* (2006) detected IRC botnet traffic in two phases: first, distinguish between IRC and non-IRC traffic by comparing existing classifiers (J48, Naïve Bayes, Bayesian network classifier); second, perform the labeling which is considered the crucial step for classifying IRC traffic as botnet or non-botnet traffic. The authors concluded that this scheme employs poor labeling criteria implemented for classification and did not provide satisfactory results.

Strayer *et al.* (2006) proposed a proactive approach to detect malicious activities by analyzing and monitoring C&C communication patterns. SLINGBot was used to send two variants for communication, TinyP2P and IRC. It was depicted that the traffic exhibited periodic behavior in both botnets. The possible advantage of this approach is that, it is independent of the communication protocol and structure used by the proposed botnets. Moreover, it does not require any prior knowledge about a certain botnet behavior.

Table 12 summarizes the machine learning tools/techniques.

Table 11 Analysis of decision tree based approaches

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
BotCop (Lu *et al.*, 2009b)	IRC botnet detection based on temporal locality	Labeling clusters is a challenging task	Temporal-frequent characteristics of network flows based on n-gram	Working under P2P environments
A framework for DNS based detection (Stalmans and Irwin, 2011)	Detection based on malicious DNS entries using the C5.0 decision tree classifier	Timely blacklist update problem	Statistical measures	Implementation of supervised learning classifiers

Table 12 Analysis of botnet detection approaches based on machine learning

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
Incremental LS-SVM learning (Chen *et al.*, 2011)	Detection of encrypted botnet communication	Focusing on the online-learning algorithm	Server IP addresses	Diverse application experiments
Spam blocking by separating end-user machines with legitimate server machines (Sanchez *et al.*, 2012)	Support vector machine (SVM)	Small dataset, undesirable for small business email servers	Machine features: OS, lexical hostname	Should be based on diverse and large datasets
Using machine learning to detect botnet (Livadas *et al.*, 2006)	A two-stage approach to detecting IRC botnets	The labeling criterion is not accurate	IP protocol flow, TCP flags, pushed packets, duration, role, etc.	Accurate labeling is required for best results
Detection based on tight C&C (Strayer *et al.*, 2006)	A proactive approach to detecting IRC botnets	Correlating the periodic traffic	Bandwidth, duration, and packet timing	

11. Group analysis based botnet detection techniques: BotSniffer (Gu *et al.*, 2008a) detects anomaly based botnet techniques particularly for HTTP and IRC botnets, and performs detection in LANs. BotSniffer tries to deceive common communication patterns of botnets, including responses and activities (for instance, sending spam emails and scanning emails) by sharing common communication content. Bots belonging to the same botnet perform identical activities, which are then identified rom the request and response patterns, and this process is called 'spatial-temporal correlation'. The concept of 'capturing synchronized botnet activities' in BotSniffer is identical to that in BotGAD (Choi *et al.*, 2009), except that BotSniffer uses a string matching concept to detect similar responses from botnets. Nevertheless, botnets can avoid communication by implementing encryption schemes or injecting random noise packets.

Gu *et al.* (2007) proposed a passive monitoring botnet detection system, BotHunter, in which IDS dialog correlation is incorporated to compare IDS events with bot infection models. The basic architecture of BotHunter is to detect malicious bot behavior at the network level. As the architecture is deployed at the network level, the stealthy bots can escape from detection by evading correlation event timings and initiate local attacks (e.g., removing files) without being noticed by end users or even without being involved in networking activities. Moreover, the authors modeled the botnet detection workflow through the following activities: target scanning, infection exploitation, binary code downloading and execution, C&C channel establishment, and outbound scanning. This scheme incorporates additional logic as it requires detecting botnets by adopting IDS-driven dialog correlation mechanisms according to specified bot infection workflow/life cycles. Therefore, emerging threats and malwares that do not conform to this model can seemingly go undetected. Apart from its disadvantages, BotHunter adopts SNORT with additional malware extensions to raise an indication/alarm when a sufficient subset of these bots has been detected. Table 13 summarizes group analysis based botnet detection tools/techniques.

12. Botnet detection techniques based on discrete Fourier transformation: Table 14 summarizes DFT based botnet detection tools/techniques. Yu X *et al.* (2010) proposed a web-based botnet detection scheme based on similarity search patterns among vast network traffic. Initially, compact feature streams were adopted from the analysis of large traffic streams for distinguishing raw traffic flows. After deep traffic analysis, an incremental discrete Fourier transform (DFT) technique was used to increase the speed of the similarity search for botnet detection. The authors performed a series of experiments to evaluate the execution time and precision of this approach. It was concluded that this method is efficient

Table 13 Botnet detection approaches based on group analysis

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
BotSniffer (Gu *et al.*, 2008a)	Network based anomaly detection, detection of C&C servers and infected machines	Unable to scan encrypted communication	Spatial temporal correlation	Support of more protocols except IRC and HTTP
BotHunter (Gu *et al.*, 2007)	Detection of malware infection through IDS-driven dialog correlation	Incorporating additional state logic, adapting to emerging threats and adversaries	-	Maximization of local dialog histories

Table 14 Summary of botnet detection approaches based on discrete Fourier transformation

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
Incremental discrete Fourier transform (Yu X *et al.*, 2010)	Detection of distributed and centralized botnet activities	-	Byte-per-packet, packet amount	Scalability of activity analysis, online clustering technique
Continuous similarity monitoring (Yu X *et al.*, 2009)	Real-time/online botnet detection framework	Difficulty in computing similarities among huge feature streams that are updated continuously	Byte-per-packet, packet amount	-

with a very low false positive rate. Additionally, the approach can be used to efficiently detect distributed and centralized botnet activities. Yu X *et al.* (2009) proposed an online botnet detection framework based on similarity search among large network traffic flows. Feature stream was introduced to characterize raw network traffic. Feature steam is the set of rules to measure the similarity ratio between request/response behaviors of network traffic. For instance, if feature stream has high similarities, the respective hosts are considered malicious bots which are added to the list of suspicious bots for further investigation. Difficulty arises in computing similarities among the huge feature streams that are updated continuously.

3.2.3 IRC based botnet detection techniques

IRC is a client/server architecture based on the application layer protocol. Within this architecture server is a chat room in which a number of clients establish connections between each other. The associated connections (channels) can be established between one-to-one or one-to-many user groups. A user can create, add, and select a channel, in order to connect to other user(s). IRC uses a default port of 194/TCP which is changeable. Moreover, password protected channels and hiding of channels are also possible for security reasons. Jing *et al.* (2009) presented the basic architecture of IRC based botnet attacks, wherein malicious activities were detected by directly monitoring IRC communication patterns. This scheme correlates common traffic patterns with additional features induced in it. Common traffic patterns that do not relate to the human standards are considered bots in the network. An anomaly-based approach (Binkley and Singh, 2006) was proposed to detect IRC specific botnets. This algorithm discovers botnet servers by combining TCP scanning with IRC detection components. This technique is not feasible for large-scale corporation networks, because a minor cipher attempt can easily defeat this approach. The correlation based algorithm (Strayer *et al.*, 2006) identifies botnet C&C servers using passive network flow analysis. It consists of three stages: filtering, classifying, and clustering. Filtering involves using filter network traffic to detect C&C. The parameters being focused on are packet size, duration, data rate, and the number of packets scanned. The classification strategy is employed to detect whether the traffic belongs to IRC or not. The parameters considered are duration, role, and the data transfer rate. The clustering strategy uses common characteristics to measure the infected IRC traffic. The variables being focused on are packet size and inter-interval time. Enormous flow induction can be used to collapse this approach by injecting massive packet- and flow-level noises into the network.

Karasaridis *et al.* (2007) proposed an anomaly-based passive algorithm, which detects botnets at the ISP level while achieving minimum false positive rates less than 2%. The basic aim of this work is to detect IRC based botnet controllers which run on randomly generated ports without knowing the signatures or taken binaries. This approach remains invisible to operators as the scheme is entirely passive. It produces false positive rates less than 2% and is deployable in large network infrastructures. Similarly, it provides botnet detection for real-time users or customers and helps identify the intensity of the botnet, size of the botnets, and characteristics of botnet activities without establishing a botnet connection. However, the critical aspect of this approach is that flow perturbation can defeat this technique. The passive monitoring botnet detection system (Goebel and Holz, 2007) tracks IRC botnet attacks using IRC nicknames as signatures. RICHI is used to monitor passive network traffic to sense IRC servers, infrequent port numbers based on ordinary and suspicious IRC nicknames. The disadvantage of this system is that it cannot detect encrypted network communication or non-IRC traffic. Table 15 summarizes IRC based botnet detection methods.

The network based system provided by Strayer *et al.* (2008) evolved from that in Strayer *et al.* (2006), which used a machine learning technique to measure botnet traffic. This process separates botnet traffic from usual network traffic and detects botnet actions by analyzing the remaining traffic flows and correlating common communication patterns which lead to botnet activities. However, this approach has the same drawback as previous research; i.e., it cannot scan encrypted network traffic. The IRC botnet detection system (Lu and Ghorbani, 2008) is based on IRC traffic characterization. This approach characterizes different flow applications by applying clustering algorithms with payload signatures to detect specific IRC application communities based on flow

Table 15 Summary of botnet detection approaches based on the IRC protocol

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
Rishi (Goebel and Holz, 2007)	IRC bot detection based on characteristics of the channel	Limitation in monitoring of protocol commands	IRC nicknames	Bots without expressions should be detected
Algorithm to detect IRC bots (Binkley and Singh, 2006)	Combining TCP scanning with an IRC detection component	Can be defeated by implementing trivial cipher schemes	IRC mesh based on IP channel names	Implementation of a mechanism to detect encrypted IRC communication
Detection based on tight C&C (Strayer *et al.*, 2006)	A proactive approach to detecting IRC botnets	Correlating the periodic traffic	Bandwidth, duration, and packet timing	-
Wide scale botnet detection (Karasaridis *et al.*, 2007)	Detecting the botnet at the ISP level	Flow perturbation can defeat this technique	Botnet controller	Integration of other forms of seed data, support for HTTP and P2P networks
Botnets detection based on IRC-community (Lu and Ghorbani, 2008)	Applying clustering algorithm with payload signatures	-	Basic flow parameters	-
Detection based on abnormal behavior of traffic (Wang Z *et al.*, 2010)	Can detect encrypted traffic, and does not require application layer information	Cannot detect IRC bots' communication on non-standard ports	NetFlow data	Detection of real-time flow

parameters. The basic theme is to separate machine-oriented IRC botnet channels from normal IRC traffic which is created by humans. An anomaly detection algorithm is applied to measure the response time of IRC communication. The basic assumption behind the response time calculation is that machines have quicker response as compared to ordinary users doing the same job. Consequently, it is possible to detect botnet communities by comparing the response time of individual communication.

One of the common difficulties that a botnet detection system faces is that the communications between C&C and its bots is quiet for certain times, as the botmaster is not always turned on to lead its bot army. If the interaction frequency between C&C and the botmaster is low enough, then it is difficult to evade botnet detection. It is a challenge for researchers to avoid this malware community even in case of small size botnets, complicated C&C architecture, and unusual interaction between the botmaster and its enemies. Wang Z *et al.* (2010) revealed that the detection of botnets was based on abnormal behavior. For this purpose, an automatic diagnostic system was designed to help analyze and dispose effectively IRC botnets. For analysis purpose, it uses NetFlow data as raw data. The basic advantage of this approach is that it does not need application layer information. In addition to that, this approach can detect encrypted traffic and find control servers and zombie masters. This approach also has some disadvantages: (1) It cannot detect IRC botnet communications on non-standard ports; (2) It cannot detect real-time flow.

3.2.4 Botnet detection based on the DNS protocol

The DNS based detection techniques use DNS (Mockapetris, 1987) information that is shared by the botnet and C&C. It is necessary to locate the botnet server to communicate with their bots. For this purpose, bots issue DNS queries to locate the C&C server. During this stage, a detection mechanism is provided to analyze DNS traffic and detect possible communication instabilities and DNS anomalies (Choi *et al.*, 2007; Villamarín-Salomón and Brustoloni, 2008). It was shown in Cranor *et al.* (2001) and Wills *et al.* (2003) that DNS queries provide information regarding botnet existence and help find the location of the C&C server. Normally bots communicate within a single administrative domain and it is easy to measure the relationship between the bots and the C&C mechanism by analyzing different domain attributes such as the lifetime of the domain, time to live (TTL) of the query, page ranking of domains, and how frequently a query is applied.

In Cranor *et al.* (2001), DNS flows were traced to identify agents including clients, DNS servers, and authoritative roots involved in DNS service provisioning. A directed graph was used wherein nodes represent IP addresses of the DNS server machines and edges represent queries generally originated by clients. Based on large-scale trace analysis, this scheme correctly identifies those agents that are involved in the DNS based botnet communication. DNS-based black list (DNSBL) (Ramachandran *et al.*, 2006) is collected from published IP addresses of the server machines or networks, with the assumption that they are involved in malicious and spamming activities. It is an attempt to grasp the botmaster address and identify its location. However, the critical aspects of this approach are that it requires an up-to-date version of the DNS-based black list and that it is difficult to design evasion techniques. In Dagon *et al.* (2006), different topological structures, the key metrics, were stated to measure the botnet phenomenon, which can be used to manage network attacks. It is shown that such metrics can be used to degrade botnet action by measuring various response techniques. DNS request rates were evaluated and DNS density rates for botnets compared. A drawback of this approach is that, by knowing the mechanisms, the botmaster can easily avoid this scheme or even suspend it from working. The botmaster can disrupt this scheme by applying massive fake DNS queries, which leads to the creation of a number of false alarms.

The botnet group activity detector, or BotGAD (Choi *et al.*, 2009), is an anomaly detection scheme based on monitoring group behavior by using DNS traffic. BotGAD provides special features to differentiate valid DNS traffic from botnet DNS queries. It enables botnet detection mechanisms on large-scale networks as well as in real-time environments. In addition, a mechanism is employed for the migration of the botnet C&C server. Since an IP header is the source to obtain DNS information, botnets with encrypted communication channels are easily traced by collecting information from the IP header. The disadvantage of this technique is that it incurs a large processing time in monitoring the vast network traffic (Feily *et al.*, 2009). Villamarín-Salomón and Brustoloni (2008) compared two different techniques (based on high DDNS query rates and abnormally recurring DDNS replies) to identify the botnet C&C server based on anomalous DDNS queries. The first technique is to look for extensive query requests for a particular domain name server. An extensive DDNS request rate is observed because botmasters apply frequent changes in their C&C servers. The second technique looks for abnormal replies from DDNS servers, to find a non-existent domain name (NXDOMAIN). Such queries result from bots that continuously locate unavailable C&C servers. A drawback of this approach is that several web servers use DNS queries with short time to live (TTL) (for example, Gmail, Yahoo, Mozilla), which can wrongly interpret those sites as botnet C&C servers. The measurement technique proposed by Abu Rajab *et al.* (2006) is used to study botnet activities by using honeypots. It measures several important behavioral and structural aspects of botnets in a very short time. A multifaceted infrastructure is employed to track multiple botnets concurrently in a given network infrastructure.

Table 16 summarizes DNS based botnet detection approaches.

3.2.5 Botnet detection based on SMTP protocols

Husna *et al.* (2008) presented a study of spamming bots which have common properties. The principal component analysis (PCA) technique was used to reveal a feature set from correlated spamming patterns. Moreover, they classified spamming patterns into different groups based on their close proximities. Zhuang *et al.* (2008) used email spam traces to detect botnet membership. Hiring the bots that participate in the same botnet email spam movement leads one to a botnet. A Hotmail web server was used for the collection of email spam traces. Different behavioral characteristics of spamming bots were analyzed, including size, number of spams sent per bot, and geographical distribution of bots. It was shown that a spam crusade is captured for only one botnet. However, as discussed in John *et al.* (2009), some spam crusades employ multiple botnets. The limitations of this approach are as follows: (1) It is difficult to distinguish botnets that are not involved in email spamming. (2) The analysis of incoming spams offers valuable information on botnet behavior as a whole; however, it lacks the ability to distinguish individual botnets and provide information regarding the latest techniques employed by spammers.

Table 16 Summary of botnet detection approaches based on the DNS protocol

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
BotGAD (Choi *et al.*, 2009)	Monitoring group behavior through DNS traffic	Increased processing time	TTL in DNS resource record, DNS querying ratio, botnet query delay, time window, size threshold	Performing well against botnet subgroups
Modeling botnet propagation using time zones (Dagon *et al.*, 2006)	Ability to predict future botnets	Focusing on centralized C&C	Time-of-release and regional-focus	Working for HTTP and P2P
Detection through analyzing DNS traffic (Villamarín-Salomón and Brustoloni, 2008)	Separating anomalous DSN traffic from normal DNS traffic	- Botnet size increase - The detection method can hardly work	TTL, CS_NS, DDNS_NS, CSAA	-
Detection based on observing group activities in DNS (Choi *et al.*, 2007)	Considering group activates in DNS traffic	Processing time is large for wide scale survey	DNS queries	Large-scale detection should be feasible
Characterizing large DNS traces using graphs (Cranor *et al.*, 2001)	Passive, active measurement, graph based analysis	Unable to handle large datasets	IP address extraction from DNS queries, graphs	Locating invalid delegates
Actively querying DNS caches (Wills *et al.*, 2003)	Botnet detection by inferring the usage pattern of applications	Does not provide perceived usage information from logs or packet traces	-	Support for other applications, log record extraction

BotGraph (Zhao *et al.*, 2009) is used to find attacks targeted at web accounts from major email service providers. This tool is used to disclose the relationship between botnet activities by building a large graph based on user-to-user relationship along with tightly coupled sub-graph components. This technique identifies stealthy isolated botnets, which are normally difficult to detect. BotGraph is considered a distributed application running on a cluster, and it is used to explore a number of techniques related to performance optimization. BotLab (John *et al.*, 2009) is used to simultaneously correlate spam emails for incoming and outgoing connections, which are collected from recognized bots in a controlled network environment. BotLab is employed to gather multiple real-time information streams which are related to a specific botnet taken from different perspectives.

The aim is to combine and analyze these various streams to produce timely, accurate, and comprehensive data about the behavior of spam botnets. As a result, this multi-perspective analysis yielded some interesting facts about spam botnet behavior. Multiple botnets participate simultaneously in a single campaign, contrary to the assumption made by prior research (Zhuang *et al.*, 2008). Similarly, 'Canadian pharmacy' is disseminated by Pushdo (Stewart, 2007), MegaD (Wikipedia, 2013b), Storm (Holz *et al.*, 2008), Kraken (Moscaritolo, 2010), and Srizbi (Keizer, 2008). This argues that the most projecting spammers' activities are performed by multiple botnets. BotLab provides a real-time platform for monitoring botnet activities and designing a network sandboxing structure, which avoids confining bots that can cause harm and disclosing new botnet variants. BotMagnifier (Stringhini *et al.*, 2011) is designed to support identifying and tracking bots that send spam. It takes an initial set of IP addresses as an input, known to be associated with spam bots, and learns their spamming behavior. This approach can effectively model spam behavior; however, the need to provide an initial IP address list is a severe shortcoming.

Table 17 shows the summary of SMTP based botnet detection techniques/tools.

3.2.6 Botnet detection based on P2P protocols

P2P traffic constitutes over 70% of the overall network traffic (Madhukar and Williamson 2006; Erman *et al.*, 2007). P2P has become a major source of illegal activities of content sharing, due to a lack of

Table 17 Summary of botnet detection approaches based on SMTP protocols

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
BOTMAGNIFIER (Stringhini *et al.*, 2011)	Detecting spamming bots	Small dataset deployed	Seed pools, transaction log	New data input, more comprehensive transaction log
Behavior analysis of spam botnets (Husna *et al.*, 2008)	Spam detection based on spammers temporal characteristics	Content type, storage time not used	Content length, time of interval, frequency of email	Considering clustering structures of telemarketing spammers
Characterizing botnets from email spam records (Zhuang *et al.*, 2008)	Botnet membership using traces of spam email	Unable to uncover botnets not involved in email spamming	Complain duration, botnet size, life span of botnets	Checking the existence of botnet in query log or ad-click log
BotLAB (John *et al.*, 2009)	A real-time botnet monitoring platform	Encrypted traffic may be overlooked	Total spam message, spam send rate, C&C protocol, C&C discovery	Detecting encrypted network traffic
BotGraph (Zhao *et al.*, 2009)	Finding attacks targeted to web accounts	-	Bot use-group size	-

control in sharing content, exchanging information, and increasing bandwidth availability to users. Many techniques have been proposed to identify and discourage this type of traffic within controlled environments (Douceur, 2002; Karagiannis *et al.*, 2004; Constantinou and Mavrommatis, 2006; Davis *et al.*, 2008; Gu *et al.*, 2008a; 2008b; Liu *et al.*, 2009; Nagaraja *et al.*, 2010; Yen and Reiter, 2010; Aviv and Haeberlen, 2011; Jian *et al.*, 2012). Table 18 summarizes the P2P based botnet detection approaches.

Common P2P applications and protocols are similar to P2P botnets; therefore, techniques employed for detecting P2P botnets are somewhat similar. Payload analysis is considered a prominent source of information for P2P applications. The parameters considered for these techniques are protocol type, port number, and the number of strings included in the packets. In Constantinou and Mavrommatis (2006), such approaches were discouraged as they are useless until enough payload information is available, and thus it is difficult to recognize unknown traffic classes.

The idea behind unavailability of payload information is that, privacy issues and legal obligations prevent network administrators from tracking and reading the actual contents of the packets (Aviv and Haeberlen, 2011). Similarly, some applications encrypt the payload information for security reasons and thus the payload information is difficult to read. Most importantly, it is unfeasible and resource intensive to classify payload information during the period of high network utilization. Moreover, unknown traffic classes such as modified or new P2P applications cannot be identified by just reading the payload information. Therefore, the above mentioned applications (modified or new P2P) use larger rather than standard port numbers to evade botnet detection mechanisms. Karagiannis *et al.* (2004) evaluated signature-based searching, which is considered the first approach to measuring the efficiencies of different signature-based techniques in dealing with packet payload.

Liu *et al.* (2009) compared the P2P networks with traditional client/server architecture through application traffic. In their findings, every node acts as a server and a client simultaneously in a P2P network environment. Therefore, P2P hosts may differ in terms of connection speed, operating system, processing capability, or network configuration. Keeping the download speed stable, a P2P host should continually initiate connections with other hosts, but because of dynamic forces involved in this system, hosts may be offline. In contrast, the connection between client/server applications has a higher success rate. BotGrep (Nagaraja *et al.*, 2010) is used to detect P2P botnets through network graph analysis. This approach focuses on the fast mixing context of the structured P2P botnet C&C graph. This approach gradually partitions the graph into slower and faster mixing pieces, ultimately narrowing it into a fast mixing component. The assumption behind this approach is that the hosts belonging to the same P2P botnet are more tightly coupled.

Table 18 Summary of botnet detection approaches based on P2P protocols

Proposed scheme	Rationale	Weakness	Relevant metric	Future direction
P2P network traffic classification (Constantinou and Mavrommatis, 2006)	An approach for P2P network identification	Does not consider application level details	Port number, packet length, packet timing	Evaluation of application specific information
Transport layer identification of P2P traffic (Karagiannis *et al.*, 2004)	A systematic methodology to identify P2P flows at the transport layer	Captured payload size, HTTP request, encryption, unidirectional traces	Identifying specific bit strings in the application-level user data	Support for flow analysis for all P2P protocols
P2P traffic identification based on support vector machine (Liu *et al.*, 2009)	Comparison of P2P systems with traditional client/server systems	Client/Server systems have higher success rates than P2P systems	Packet length, remote hosts' discreteness, connection response rate	Implementation at the server level
BotGrep (Nagaraja *et al.*, 2010)	Finding P2P bots with structured graph analysis	Content evaluation	Processing costs, bandwidth overhead	Observation of more fine-grained properties of communication patterns
Sybil attacks to mitigation the Storm botnet (Davis *et al.*, 2008)	Effects of Sybil attacks on botnet C&C	Requirement of QoS services be explored	Sybil population size, size of bots' peer-list	Provision of robustness and resilience measures
Detection based on P2P file sharing (Yen and Reiter, 2010)	Separation of P2P bots from P2P file-sharing hosts	-	Volume, peer churn, and human-driven versus machine-driven	-
P2P botnet detection based on network streams analysis (Liu *et al.*, 2010)	Detection algorithm, clustering algorithm, similarity detection algorithm	Results taken from the LAN simulation environment, lengthy process	Polymorphic, undiscovered, cryptographic channel	Testing for Sinit, Phatbot P2P botnets
An evaluation model of botnet based on P2P (Jian *et al.*, 2012)	Evaluation model based on some botnets' concrete parameters	Simulation study	Stealthiness, effectiveness, efficiency, and robustness	Application on real-world botnets
PeerPress: using enemies' P2P strength against them (Xu *et al.*, 2012)	P2P botnet detection by exploiting the enemies' strength against them	Difficulty in detecting advanced encrypted communication	Portprint extraction	

Sybil attack (Douceur, 2002) targets the systems with the assumption of multiple identities due to the lack of any certification authority in P2P networks. In a Sybil attack, the attacker disrupts the known system in the P2P network by creating several fake entities with similar identifications to cause a large influence on the network. A target's system vulnerability in a Sybil attack totally depends on how inexpensive the generation of identities can be and how well the target system treats those entities. This technique can access Storm botnets (Moscaritolo, 2010) which measure a series of features. Davis *et al.* (2008) developed a technique based on previous findings, which generates a false positive rate during Storm botnets and is able to send unrelated commands to subvert the C&C channel. Constantinou and Mavrommatis (2006) proposed a novel technique for detection of P2P botnets, based on the primary characteristics of P2P protocols rather than application-specific details. The characteristics include a vast network diameter and various entities working as both client and server. Yen and Reiter (2010) differentiated hosts performing legal P2P activities from P2P bots. They analyzed the network flows generated by Argus and found characteristics related to the connection time between P2P networks, flow volume, and behavior of hosts (machine- or human-oriented). The Storm botnet and Nugache botnet can be identified in 87.5% and 34% of their occurrences, respectively.

BotMiner (Gu *et al.*, 2008b) employs data mining concept to detect botnet C&C. BotMiner is the improved version of BotSniffer (Gu *et al.*, 2008a). BotMiner collects similar communication patterns for malicious traffic and performs cross cluster correlation to detect the hosts that belong to the same communication patterns and malicious activities. BotMiner is independent of the botnet structure and protocol. Furthermore, it can detect real-world botnets, including HTTP-, IRC-, and P2P-based botnets, with low false-positive rates. According to Liu *et al.* (2010), P2P botnet detection methods can be classified as Detection based on the Protocol Feature codes (DoPF) and Detection based on the Network Streams (DoNS). DoPF seems to have less efficiency than DoNS. Various DoNS algorithms have been published. Karagiannis *et al.* (2003) suggested one that distinguishes P2P streams from the size of the datagram. Furthermore, Karagiannis *et al.* (2004) and Zeidanloo *et al.* (2010) suggested a combined P2P application detection method using the features of the network streams. Zhou *et al.* (2006) introduced a detection method based on the rate of the successful connections between network flows. After performing extensive analysis on the actions of nodes in functions, networks, and applications, Karagiannis *et al.* (2005) proposed an algorithm to identify the P2P nodes.

Jian *et al.* (2012) constructed a comprehensive evaluation model which can evaluate the performance of P2P botnets in different perspectives: (1) effectiveness, (2) stealthiness, (3) efficiency, and (4) robustness. By analyzing these four important simulation indexes, the authors provided a detailed calculation formula to evaluate the relationship between botnets and the intensity of their attack. The basic problem with this approach is the lack of this implementation model in real-world scenarios.

Xu *et al.* (2012) presented a P2P passive botnet detection technique which can effectively identify P2P malware codes by exploiting the botmaster strength against them. The two-phase detection framework is robust in host-level dynamic binary code analysis with network-level probing, based on the assumption that usually a P2P mechanism has a remotely controllable built-in architecture (through opening some ports for binary code access), which can be exploited to observe the malicious behavior of the nodes. Besides the effectiveness of this approach, advanced encryption and certificate based authentication may also evade detection. Moreover, variations in the port binding delay may make detection difficult for this scheme.

3.2.7 Web based botnet detection techniques

Users rely on web surfing and Internet services which result in a new type of botnet attack called the HTTP (or web) based botnet. It is hard to identify and locate such a type of botnet attack because the communication traffic or attack emerges into the normal Internet traffic. Web bots require regular connection with the server and specifically invariant page size, whereas a casual user's web traffic requires randomness on web page size and visit time. Therefore, HTTP botnet attacks are serious because the hacker takes advantage of the HTTP connections to make the malicious traffic be encapsulated within the huge amount of standard traffic and thus difficult to detect. Wang B *et al.* (2010) proposed an architecture for detection of web-based botnets in a supervised network through modeling behavioral characteristics of bots. After investigation of a large number of web-based bots, it is concluded that for the connections made with C&C communications and various other activities, the bot behaves in a similar fashion in terms of statistical meaning. Similar connections appear periodically despite the fact that the parameters are different. However, this technique depends on neither bot group activities nor traffic payload information. Moreover, this approach detects bots with encrypted web-based communication and also as a single infected bot in a managed network. It works best in detecting web-based botnet attacks, showing a low false-positive rate.

Wang B *et al.* (2010) compared two types of botnets, including the web-based botnets using pull-based infection and the traditional botnets. Moreover, the IRC and HTTP protocols can control botnets created by web malware (Stringhini *et al.*, 2011). A more recent HTTP based botnet detection technique (Chen *et al.*, 2013) is devised, based on the concept of 'fast-flux domains'. The proposed web-based botnet detection technique analyzes traffic flow to determine botnets that use HTTP as the C&C channel or employ fast-flux network domains for concealing. The authors examined the proposed model on both a real network environment and a virtual testbed and verified that the proposed scheme can effectively diagnose HTTP or web based botnets.

4 Future trends of the botnet phenomenon

Security concerns are steadily growing as a result of the integration of different computing and communication technologies, globalization, and the mass market economy of the world. Security is extremely important in information technology which needs to adapt to the rapid changes in the industry and cope with the lessons learned through several high profile exploits of the server and data vulnerabilities. Botnets operate like a distributed network and are one of the most dangerous threats on the web for the modern distributed computing models. In this section we discuss the future trends of the botnet phenomenon.

4.1 Social botnets

The primary concern for the botmaster and cybercriminal is to capture a huge audience while remaining hidden from them; therefore, they try to exploit social media sites such as Facebook and Twitter. Message propagation is quick in social networks and people trust the links they receive. Social media networks provide a number of services such as banking transactions, wall management, event logs, Q&A, online forums, news feeds, messages and inboxes, IPv6 support, friend lists, and gaming. Eventually, exploiting loopholes within these applications can lead to the advent of highly sophisticated fraud schemes. Social networks are becoming the great challenge for evasion of botnets owing to the tight relationships between social networks and botnets. Bot recruitment can be considered an active portion for social networks; for instance, malicious code can be shared to contaminate the victim's machine, which can then treat them as a zombie and also host C&C architecture. From all these aspects, one of the most important issues lies in the difficulty in detecting social networks as a botnet. For example, it is common for the botnet author to infect machines by creating fake credentials to start encrypted communications to target machines through various social media networks. Svelta malware (Emre, 2011) is considered the major source of malware distribution in social media networks.

Botnet Butterfly (Wikipedia, 2013c) or Mariposa is considered one of the profitable botnets in modern history. In 2008, a Butterfly botnet infected and damaged 12 million PCs worldwide. It was originally designed to perform illegal activities including phishing and spamming services, DDoS attacks, and stealing of important and sensitive information. A number of organizations such as the FBI are trying to develop new tools to inspect fraudulent activities which abuse social media networks to throttle a large number of victims (FBI, 2012). Recently, Facebook assisted the FBI in busting such cybercriminals. As a result, the US Department of Justice notified and arrested 10 individuals which were involved in spreading Yahoo's botnet attack. Such attacks infected nearly 11 million computers all over the world and resulted in a financial loss of over 850 million USD. This report also states that the botnet affected Facebook users from October 2010 to December 2012. Similarly, in 2012 a virus named 'Rammit' targeted Facebook subscribers and stole the passwords and usernames of over 45 000 subscribers around the world—most of them were from UK (69%) and France (27%) (Raff, 2012). To prevent the propagation of malicious agents, it is fundamental that users should adopt proper behavior and use updated security defense systems.

4.2 Mobile botnets

In the rapidly growing mobile computing world, mobile botnets are a serious threat to mobile phone devices such as smartphones. The aim of this attack is to gain access to the resources and contents of the mobile user's device and send control instructions to the botnet initiator. The hackers take advantage of the open exploited area of mobile devices to gain unauthorized access over the compromised mobile devices. Eventually, the hacker's goal is to perform malicious and unauthorized activities including illegal phone calls, accessing a control panel, sending emails, initialization of worm code, and unauthorized file or photo access (Cui *et al.*, 2011). Table 19 shows some of the mobile botnet attacks.

Recently, a number of botnets have been evolved that can disrupt the performance of the mobile device. For instance, ZeuS (Schwartz, 2012) is a botnet that focuses on Blackberry, Symbian, and Windows platform users, and the DreamDroid botnet (Tung, 2011) affects Android based devices. Similarly, IKee.B (The H Security, 2007) is a botnet that is used to scan IP addresses of iPhones, whereas Android BMaster and TigerBot specifically target Android application frameworks.

Table 19 Possible mobile botnet attacks

Attack type	Description
Sending email	A mobile bot Weldac was designed to send emails without being noticed by the mobile user
Sending MMS/SMS	An infected mobile may send MMS/SMS to service providers or to a wide range of subscribers. An SMS based heterogeneous mobile botnet (Ahmed *et al.*, 2013) was created to perform a similar task
Victim selection	Victims/bot enemies can be selected by the botmaster from the contact list or address book of infected mobile devices
Mobile voting system	A botmaster can dismiss recently evolved mobile voting services
Charity service	Giving money to charity organizations using mobile services may be exploited by the mobile botnet
Spyware	Infected mobiles can be treated as a spyware to collect personal information of subscribers
Privacy issue	Privacy issues may arise in mobile networks when personal information (for instance, credit card number or financial information) is stolen while the mobile user is interacting with response servers (Ahmed *et al.*, 2013)

Andbot (Cui *et al.*, 2011) is a mobile bot which employs URL (uniform resource locator) flux. It is considered a stealthy, low-cost, and resilient bot, which attracts botmasters to use illegal activities in the mobile environment. This botnet uses microblogs to send malicious commands. Andbot can be easily implemented on smartphones and could be sustained for a long time without being noticed or detected. Andbot integrates several other schemes to be efficient and stealthy. The cloud based push-styled mobile botnet (Zhao *et al.*, 2012) is a new type of botnet in the mobile environment; it uses push-based notification services to disseminate the commands. A novel C&C channel is presented using a cloud-to-device-messaging (C2DM) service which is provided by Google for Android platforms. C2DM is shown to be stealthy in terms of command traffic, requires less power consumption, has optimized bandwidth utilization, and is controllable in the efficient transformation of commands to all bots. Similarly, epidemic mobile malware is a new terrifying threat for mobile users (Szongott *et al.*, 2012), which disseminates rapidly in smartphones. The malware affected older versions of iOS; however, epidemic mobile malware is still a predominant threat for mobile users. Mobile botnets are a relatively new research domain, in which there are a number of problems that need to be addressed. The detection, analysis, and mitigation of mobile botnets have become hot research topics.

4.3 Botnets to botclouds

Botnet rival is the power of cloud computing platforms. These 'dark' clouds, controlled by cybercriminals, are designed to silently infect networks. Left undetected, botnets borrow the network to serve malicious business interests. Therefore, botnets are not restricted to harming traditional network machines; the sophistication of the botnet enables the control of cloud services as well. Similarly, the botmaster's responsibility is changing with the growing need for technology. As a result, a large number of machines are arranged from a cloud service provider (CSP) and the bot is configured on each machine which works to ensure future malicious activities on the cloud. The advantages of cloud based botnets or BotClouds (Proffitt, 2012) over traditional botnets (Jing *et al.*, 2009) are as follows: (1) They require less time to converge; (2) A BotCloud is always accessible; (3) They maximize the utilization of the cloud resources. As previously mentioned, there are two methods for botnet detection, honeypots and IDS. Deployment of these techniques on clouds is complicated. For instance, porting honeypots on a cloud requires service-level agreements (SLAs) with CSPs to monitor the activity logs of machines used by customers.

Questions could arise from the security perspective of hardware resources, as this intensive hardware is the sole property of organizations (Ruiter and Warnier, 2011). Similarly, a number of log monitoring systems are lacking in providing guarantees to differentiate legitimate and illegitimate activities. Similarly, implementing IDS on individual cloud machines is not a straightforward task. IDS algorithms are suitable for a safe baseline, which is considered a 'normal' network activity. When a new activity arises, it is compared with the baseline activity and declared as a malicious activity if found suspicious. So, it is an impractical task to develop this safe baseline for individual cloud users, as implementing a comparison for each transaction is time consuming and resource intensive. Comazzetto (2011) proposed security gateways that offer comprehensive unified threat management systems against threats of the botnet. Botnet activities on clouds are controlled by the following three simple rules: (1) The OS and application programs should be patched and updated; (2) The deployment of an effective gateway defense solution on the access layer is necessary to prevent bots from entering personal computers; (3) Gradually testing the organization's workstations and servers helps reduce botnet activities on the cloud.

4.4 Latest botnet attacks

1. StealRat botnet: Recently, hackers have designed a sophisticated botnet called 'StealRat' (Torre, 2013), which is capable of evading organization's advanced anti-spam defenses. StealRat uses specialized techniques to hide the malware used in the scam. The botnet can bypass most organization's cyber defense systems by minimizing interaction between the campaign's server and spam messages. The tactic behind the StealRat botnet is to hide its operations in three layers, an infected machine and two compromised websites. Recently, Trend Micro (Torre, 2013) estimated that the attackers were using approximately 85000 different domains or IP addresses to disseminate spam to seven million selected email addresses. The discovery of the StealRat botnet comes from a large-scale evaluation of cybercriminals' attack techniques.

2. Citadel botnet: Another recently evolved malware called the 'Citadel botnet' uses malicious codes to not only harm personal computers but also evade from the malware recognition process of various anti-virus/anti-malware software. During the preliminary investigation (Constantin, 2013), the researchers found that the Citadel botnet restricts access to many legitimate anti-malware sites, so that people cannot access these anti-malware sites to scan their machines for possible attacks. According to Schwartz (2013), a gang is behind the Citadel botnets trying to spread malware by distributing pirated Windows XP in which the malware is pre-configured. A joint effort has been made to defeat the Citadel botnet (Schwartz, 2013) involving not only Microsoft and the FBI but also the US Marshals Service.

3. Andromeda botnet: The Andromeda botnet (Trend Micro, 2013), first marketed in late 2011, has recently re-emerged. This threat arrives through a prominent way in which spammed messages with malicious code attachments or links are forwarded to compromised websites hosting BlackHole Exploit Kit (BHEK) code. The Andromeda botnet itself is a highly modular program which incorporates various modules including (1) form grabbers, (2) SOCKS4 proxy module, (3) keyloggers, and (4) rootkits. Moreover, as is typical of backdoors, it can download and execute other files like ZeuS, as well as update and remove itself if needed.

4. Massive attacks on WordPress targeting the 'admin' password: WordPress is a popular open source blogging tool to make websites more powerful. A recent web-based brute force attack (Press, 2013) was launched against the admin account of WordPress. According to Gilbertson (2013), there is nothing new in this attack. The only thing that makes this attack different and potent is that the attackers have more than 90000 unique IP addresses to make this attack more devastating.

5. Android master key vulnerability: Another serious vulnerability was seen recently on 99% of all Android based devices that can connect to a company's network to gain access to confidential data related to the organization's policies (Forristal, 2013). The Bluebox security team has recently discovered this vulnerability in Android's security model, which allows an attacker to modify application (.apk) code without damaging an application's encrypted signatures that lead to a legitimate application into a malicious Trojan, with the ability to go completely unnoticed by the phone, application store, or even end user.

Android master key vulnerability is risky for both enterprises and individuals (as a malicious application can gain access to an enterprise database or even individual data). This threat is multiplied when considering applications designed by the device vendors (e.g., Samsung, HTC, LG, Motorola) or third parties that work in collaboration with the device manufacturer (e.g., Cisco with AnyConnect VPN) that grant special privileges within Android, specifically granting access to a system's UID.

5 Open issues in the botnet detection phenomenon

Botnets have become a global phenomenon and the botmaster has the responsibility to capture thousands of vulnerable hosts in domains around the world. There are several challenges that surround the study of botnets and botnet detection. The following are some of the key issues concerning botnet detection on a wide scale:

One of the important factors in recognizing the intensity of botnet threats is to assess the affected botnet span. Existing detection approaches normally lack accuracy in measuring the size of botnets and the numbers generated are acceptable only to a very limited degree. Additionally, statements about botnet sizes seldom provide a clear mention about the scientific basis used by different approaches or the applied measurement methodologies.

The administrative domains consider their detailed information a business secret which is not shared with outsiders or researchers. Researchers have access to a small portion of that data by signing a usage agreement which is manually defined for each domain separately. Further, in the context of cybercrimes, the current legislative frameworks of various European states and their national diversity are the major factors in recognizing the efficiency of the fight against botnets. The applicability of promising detection and mitigation approaches is also restricted through certain conflicts between data protection laws and laws that govern the secure operation of IT services (Plohmann *et al.*, 2011). Finally, working processes increase the reaction time to the extent that they can be evaded with little effort by criminal individuals, capitalizing on the ease with which botnets can be configured.

Data traces are effective for differentiating illegal activities from legitimate traffic patterns; however, the perception behind these data traces is that, these may include sensitive information for a given administrative domain. Therefore, data traces are handled and manipulated carefully even within the same organization. Moreover, it is difficult to obtain real traces, whereas researchers require contents for evaluating the performance of their systems on a small set of data traces, which is a challenging task because heterogeneity (differences in architectural design, software, hardware, etc.) on the Internet is not clearly defined for many datasets. Alternatively, many synthetic traces (Weigle *et al.*, 2006; Vishwanath and Vahdat, 2009) and various botnet emulators (Lee, 2009) are available but still have their own limitations with respect to validity of results and potential biasness. Similarly, researchers face difficulty in comparing their results with previously published benchmarks, because the datasets to a full extent are not easily accessible for the researcher community.

With the growing increase in computing proficiencies and Internet usage (e.g., WiFi, 3G, and GPRS) for mobile devices including smartphones and handheld devices, another threat has become extremely frightening in this field, i.e., the botnet phenomenon in mobile devices and its detection. Research is at the initial stage in this area. Traynor *et al.* (2009) found loopholes in wireless and cellular networks and showed how to degrade a wireless network by incorporating attacks on cellular networks. Mulliner and Seifert (2010) and Fogarty (2011) described in detail two cellular botnets known as Andbot and IBot, including the distribution of models, C&C channels, and malicious commands. The mobile botnet detection mechanism is restricted because of several factors, including: (1) limited battery power, (2) limited bandwidth, (3) saturated phone service, (4) GPS data, (5) SMS messages, (6) tracking other mobile devices, (7) private IP addresses, etc. Fig. 6 shows the challenges inherited from traditional botnets to mobile botnets.

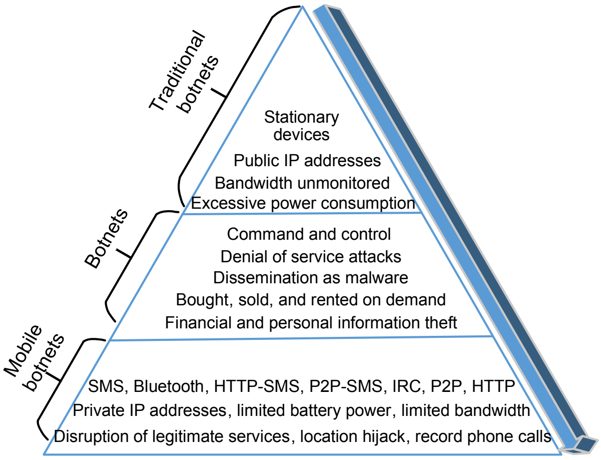
Traditional botnets
Stationary devices
Public IP addresses
Bandwidth unmonitored
Excessive power consumption
Botnets
Command and control
Denial of service attacks
Dissemination as malware
Bought, sold, and rented on demand
Financial and personal information theft
Mobile botnets
SMS, Bluetooth, HTTP-SMS, P2P-SMS, IRC, P2P, HTTP
Private IP addresses, limited battery power, limited bandwidth
Disruption of legitimate services, location hijack, record phone calls

Fig. 6 Challenges inherited from traditional botnets to mobile botnets

Another problem confronted by cellular networks is that, although cellular companies try to protect their customers from outside malicious interferences, because of dynamic network configurations (address assigned by network administrators at connection setup time) of mobile nodes, most of the time it is difficult to detect vulnerabilities, thus making the provider's efforts impractical. Moreover, in the area of mobile communication, there is a lack of certificate provided by companies for user applications. Therefore, it is easy to find backdoors in these applications, allowing the applications to propagate botnet. The situation may worsen with the emergence of mobile payment systems, which are aimed at replacing existing credit cards.

The global botnet threat is best confronted by close international cooperation between governments and technically oriented and legislative institutions. For an efficient supranational detection and mitigation strategy to work, liaison between stakeholders must be strengthened and intensified through political support and will.

In this context, the standardization of processes for information exchange plays a vital role. This includes documentation of reports about identified activities/threats, incidents, and strong evidence against criminal individuals, probably leading to their arrest, as well as built-in mechanisms for maintaining the confidentiality of shared information and establishing the trustworthiness of its sources.

6 Conclusions

A comprehensive review of the latest state-of-the-art for botnet detection techniques is presented to figure out the trends of previous and current research and the issues in the botnet detection phenomenon. A thematic taxonomy is proposed for the classification of botnet detection techniques, and the implications and critical aspects are highlighted through qualitative analysis of such techniques. We also discuss recent trends towards botnets that are emerging with new technologies, and the open challenges in botnet detection are highlighted for future research. Botnets have become a global threat; therefore, it is necessary for different stakeholders (network personnel, administrative entities, etc.) to take collaborative actions to eliminate this harmful hazard. Similarly, it is important to negotiate on possible international legislative issues and establish global policies to systematically address the threats of the botnet phenomenon. Currently, botnet detection techniques are employed to collect flow information from bots to depict their behavior and their detection mechanism. However, a number of challenges still persist in the area of botnet detection. For instance, researchers face the problem of validating their proposals in real network environments using existing data. Creating trace repositories has already been implemented with little success; however, data access is still restricted within various administrative domains. Therefore, research challenges still exist for distributed and collaborative global botnet countermeasures and strong Inter-AS communication systems.

References

Abu Rajab, M., Zarfoss, J., Monrose, F., *et al.*, 2006. Amultifaceted approach to understanding the botnet phenomenon. Proc. 6th ACM SIGCOMM Conf. on Internet Measurement, p.41-52.

Ahmed, R., Dharaskar, R.V., Thakare, V.M., 2013. Efficient generalized forensics framework for extraction and documentation of evidence from mobile devices. *Int. J. Enhanced Res. Manag. Comput. Appl.*, **2**(1):1-7.

Aviv, A.J., Haeberlen, A., 2011. Challenges in experimenting with botnet detection systems. USENIX 4th CSET Workshop, p.1-8.

Bailey, M., Cooke, E., Jahanian, F., *et al.*, 2009. A survey of botnet technology and defenses. IEEE Cybersecurity Applications & Technology Conf. for Homeland Security, p.299-304. [doi:10.1109/CATCH.2009.40]

Barford, P., Yegneswaran, V., 2007. An inside look at botnets. *In*: Malware Detection. Springer, p.171-191. [doi:10.1007/978-0-387-44599-1_8]

Barsamian, A.V., 2009. Network Characterization for Botnet Detection Using Statistical-Behavioral Methods. Master Thesis, Dartmouth College.

Bauer, J., van Eeten, M., Chattopadhyay, T., 2008. ITU Study on the Financial Aspects of Network Security: Malware and Spam. Final Report, ICT Applications and Cybersecurity Division, International Telecommunication Union.

BBC, 2008. Technology | Spam on Rise after Brief Reprieve. BBC News. Available from http://news.bbc.co.uk/2/hi/technology/7749835.stm [Accessed on Dec. 3, 2013].

Bethencourt, J., Franklin, J., Vernon, M., 2005. Mapping Internet sensors with probe response attacks. Proc. 14th USENIX Security Symp., p.193-208.

Bhuyan, M., Bhattacharyya, D., Kalita, J., 2013. Network anomaly detection: methods, systems and tools. *IEEE*

Commun. Surv. Tutor., **16**(1):1-24.

Binkley, J.R., Singh, S., 2006. An algorithm for anomaly-based botnet detection. Proc. USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop, p.43-48.

Bu, Z., Bueno, P., Kashyap, R., *et al.*, 2010. The New Era of Botnets. Available from http://www.mcafee.com/in/resources/white-papers/wp-new-era-of-botnets.pdf [Accessed on Sept. 9, 2013].

Cai, T., Zou, F., 2012. Detecting HTTP botnet with clustering network traffic. IEEE 8th Int. Conf. on Wireless Communications, Networking and Mobile Computing, p.1-7.

Ceron, J.M., Granville, L.Z., Tarouco, L.M., 2008. Uma arquitetura baseada em assinaturas para mitiga cao de botnets. *In*: X Simposio Brasileiro em Seguran ca da Informa cao e de Sistemas Computacionais (SBSeg), p.105-118 (in Portuguese).

Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: a survey. *ACM Comput. Surv.*, **41**(3):1-58.

Chang, S., Daniels, T.E., 2009. P2P botnet detection using behavior clustering & statistical tests. Proc. 2nd ACM Workshop on Security and Artificial Intelligence, p.23-30. [doi:10.1145/1654988.1654996]

Chen, C.M., Huang, M.Z., Ou, Y.H., 2013. Detecting web-based botnets with fast-flux domains. Advances in Intelligent Systems and Applications, Volume 2. Springer, p.79-89. [doi:10.1007/978-3-642-35473-1_9]

Chen, F., Ranjan, S., Tan, P., 2011. Detecting bots via incremental LS-SVM learning with dynamic feature adaptation. Proc. 17th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, p.386-394.

Choi, H., Lee, H., Lee, H., *et al.*, 2007. Botnet detection by monitoring group activities in DNS traffic. 7th IEEE Int. Conf. on Computer and Information Technology, p.715-720. [doi:10.1109/CIT.2007.90]

Choi, H., Lee, H., Kim, H., 2009. BotGAD: detecting botnets by capturing group activities in network traffic. Proc. 4th Int. ICST Conf. on Communication System Software and Middleware, p.1-8. [doi:10.1145/1621890.1621893]

Choi, Y.H., Li, L., Liu, P., *et al.*, 2010. Worm virulence estimation for the containment of local worm outbreak. *Comput. & Secur.*, **29**(1):104-123. [doi:10.1016/j.cose.2009.07.002]

Comazzetto, A., 2011. Botnets: the Dark Side of Cloud Computing. Technical Report, Bostan, USA.

Constantin, L., 2013. Microsoft: Almost 90 Percent of Citadel Botnets in the World Disrupted in June. Available from http://www.pcworld.com/article/2045282/microsoft-almost-90-percent-of-citadel-botnets-in-the-world-disrupted-in-june.html [Accessed on July 6, 2013].

Constantinou, F., Mavrommatis, P., 2006. Identifying known and unknown peer-to-peer traffic. 5th IEEE Int. Symp. on Network Computing and Applications, p.93-102. [doi:10.1109/NCA.2006.34]

Cooke, E., Jahanian, F., McPherson, D., 2005. The zombie roundup: understanding, detecting, and disrupting botnets. Proc. USENIX SRUTI Workshop, p.44.

Coskun, B., Dietrich, S., Memon, N., 2010. Friends of an enemy: identifying local members of peer-to-peer botnets using mutual contacts. Proc. 26th Annual Computer Security Applications Conf., p.131-140. [doi:10.1145/1920261.1920283]

Cranor, C.D., Gansner, E., Krishnamurthy, B., *et al.*, 2001. Characterizing large DNS traces using graphs. Proc. 1st ACM SIGCOMM Workshop on Internet Measurement, p.55-67. [doi:10.1145/505202.505210]

Creech, G., Hu, J., 2013. A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns. *IEEE Trans. Comput.*, **1**(1):1-23.

Cremonini, M., Riccardi, M., 2009. The Dorothy project: an open botnet analysis framework for automatic tracking and activity visualization. IEEE European Conf. on Computer Network Defense, p.52-54.

Crowfoot, S., 2012. Trojan.Bredolab Spreading in PDF Download. Available from http://www.iceni.com/blog/trojan-bredolab-spreading-in-pdf-download/ [Accessed on Oct. 4, 2014].

Cui, X., Fang, B., Yin, L., Xiang, C., *et al.*, 2011. Andbot: towards advanced mobile botnets. Proc. 4th USENIX Conf. on Large-Scale Exploits and Emergent Threats, p.11.

Dagon, D., Zou, C.C., Lee, W., 2006. Modeling botnet propagation using time zones. NDSS, **6**:2-13.

Dagon, D., Gu, G., Lee, C.P., *et al.*, 2007. A taxonomy of botnet structures. IEEE 23rd Annual Computer Security Applications Conf., p.325-339.

Danchev, D., 2009. Research: Small DIY Botnets Prevalent in Enterprise Networks. Available from http://www.zdnet.com/blog/security/research-small-diy-botnets-prevalent-in-enterprise-networks/4485[Accessed on Oct. 13, 2014].

Davis, C.R., Fernandez, J.M., Neville, S., *et al.*, 2008. Sybil attacks as a mitigation strategy against the Storm botnet. IEEE 3rd Int. Conf. on Malicious and Unwanted Software, p.32-40.

di Pietro, R., Mancini, L.V., 2008. Intrusion Detection Systems. Springer.

Douceur, J.R., 2002. The sybil attack. *In*: Peer-to-Peer Systems. Springer Berlin Heidelberg, p.251-260. [doi:10.1007/3-540-45748-8_24]

Emre, Y., 2011. A literature survey about recent botnet trends, p.1-14.

Erman, J., Mahanti, A., Arlitt, M., *et al.*, 2007. Identifying and discriminating between web and peer-to-peer traffic in the network core. Proc. 16th Int. Conf. on World Wide Web, p.883-892. [doi:10.1145/1242572.1242692]

Falliere, N., 2011. Sality: Story of a Peer-to-Peer Viral Network. Symantic Security Response, Technical Report.

Falliere, N., Murchu, L.O., Chien, E., 2011. W32.Stuxnet Dossier, Version 1.4. White Paper, Symantec Security Response.

FBI, 2012. FBI, International Law Enforcement Disrupt International Organized Cyber Crime Ring Related to Butterfly Botnet.

Feily, M., Shahrestani, A., Ramadass, S., 2009. A survey of botnet and botnet detection. IEEE 3rd Int. Conf. on Emerging Security Information, Systems and Technologies, p.268-273.

Fogarty, K., 2011. Just What We Need: Malware to Slave Your Android to a Botnet. IT World. Available from http://www.itworld.com/article/2732959/mobile/just-what-we-need--malware-to-slave-your-android-to-a-botnet.html [Accessed on June 20, 2014].

Forristal, J., 2013. Uncovering Android Master Key That Makes 99% of Devices Vulnerable. Available from https://bluebox.com/technical/uncovering-android-master-key-that-makes-99-of-devices-vulnerable/ [Accessed on Oct. 4, 2014].

Fossi, M., Egan, G., Haley, K., *et al.*, 2011. Symantec Internet Security Threat Report Trends for 2010. Symantec Internet Security Threat Report, Volume 16, p.1-20.

Francia, R., 2007. Storm Worm Network Shrinks to About One-Tenth of Its Former Size. Tech. Blorge. Com., p.10-21.

François, J., Wang, S., Engel, T., 2011. BotTrack: tracking botnets using NetFlow and PageRank. NETWORKING, p.1-14.

Freiling, F.C., Holz, T., Wicherski, G., 2005. Botnet tracking: exploring a root-cause methodology to prevent distributed denial-of-service attacks. Springer Berlin Heidelberg, p.319-335.

Ge, L., Liu, H., Zhang, D., *et al.*, 2012. On effective sampling techniques for host-based intrusion detection in MANET. IEEE Military Communications Conf., p.1-6.

Gilbertson, S., 2013. Massive WordPress Attack Targets Weak Admin Passwords. Available from http://www.webmonkey.com/2013/04/massive-wordpress-attack-targets-weak-admin-passwords [Accessed on Sept. 8, 2013].

Goebel, J., Holz, T., 2007. Rishi: identify bot contaminated hosts by irc nickname evaluation. Proc. 1st Conf. on 1st Workshop on Hot Topics in Understanding Botnets, p.1-12.

Goodin, D., 2008. Botnet Sics Zombie Soldiers on Gimpy Websites. Available from http://www.theregister.co.uk/2008/05/14/asprox_attacks_websites/ [Accessed on June 6, 2013].

Goodin, D., 2010. Waledac Botnet 'Decimated' by MS Takedown. Available from http://www.theregister.co.uk/2010/03/16/waledac_takedown_success/ [Accessed on June 8, 2013].

Grizzard, J., Sharma, V., Nunnery, C., 2007. Peer-to-peer botnets: overview and case study. Proc. 1st USENIX Workshop on Hot Topics in Understanding Botnets, p.1.

Gu, G., Porras, P., Yegneswaran, V., *et al.*, 2007. Bothunter: detecting malware infection through IDS-driven dialog correlation. Proc. 16th USENIX Security Symp., p.167-182.

Gu, G., Zhang, J., Lee, W., 2008a. BotSniffer: detecting botnet command and control channels in network traffic. Proc. 15th Annual Network and Distributed System Security Symp., p.2-19.

Gu, G., Perdisci, R., Zhang, J., *et al.*, 2008b. BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection. USENIX Security Symp., p.139-154.

Gu, G., Yegneswaran, V., Porras, P., *et al.*, 2009. Active botnet probing to identify obscure command and control channels. IEEE Annual Computer Security Applications Conf., p.241-253.

Ha, D.T., Yan, G., Eidenbenz, S., *et al.*, 2009. On the effectiveness of structural detection and defense against P2P-based botnets. IEEE/IFIP Int. Conf. on Dependable Systems & Networks, p.297-306.

Holz, T., Steiner, M., Dahl, F., *et al.*, 2008. Measurements and mitigation of peer-to-peer-based botnets: a case study on Storm worm. LEET, **8**(1):1-9.

Huang, S.Y., Mao, C.H., Lee, H.M., 2010. Fast-flux service network detection based on spatial snapshot mechanism for delay-free detection. Proc. 5th ACM Symp. on Information, Computer and Communications Security, p.101-111.

Husna, H., Phithakkitnukoon, S., Palla, S., *et al.*, 2008. Behavior analysis of spam botnets. IEEE 3rd Int. Conf. on Communication Systems Software and Middleware and Workshops, p.246-253.

Ianelli, N., Hackworth, A., 2005. Botnets as a vehicle for online crime. CERT Coordination Center, **1**(1):28.

Iliofotou, M., Pappu, P., Faloutsos, M., *et al.*, 2007. Network monitoring using traffic dispersion graphs (TDGS). Proc. 7th ACM SIGCOMM Conf. on Internet Measurement, p.315-320. [doi:10.1145/1298306.1298349]

Jackson, K., 2008. New Massive Botnet Twice the Size of Storm. Available from http://www.darkreading.com/security/news/211201307 [Accessed on May 5, 2014].

Janssen, C., 2011. Global Threat Bot (GTbot). Available from http://www.techopedia.com/definition/59/global-threat-bot-gtbot [Accessed on May 6, 2014].

Jelasity, M., Bilicki, V., Kasza, M., 2011. Modeling network-level impacts of P2P flows. 19th IEEE Euromicro Int. Conf. on Parallel, Distributed and Network-Based Processing, p.590-594.

Jian, G., Zheng, K., Yang, Y., *et al.*, 2012. An evaluation model of botnet based on peer to peer. IEEE 4th Int. Conf. on Computational Intelligence and Communication Networks, p.925-929.

Jiang, N., Cao, J., Jin, Y., *et al.*, 2010. Identifying suspicious activities through DNS failure graph analysis. 18th IEEE Int. Conf. on Network Protocols, p.144-153. [doi:10.1109/ICNP.2010.5762763]

Jing, L., Yang, X., Kaveh, G., *et al.*, 2009. Botnet: classification, attacks, detection, tracing, and preventive measures. *EURASIP J. Wirel. Commun. Network.*, **2009**: 1-11.

John, J.P., Moshchuk, A., Gribble, S.D., *et al.*, 2009. Studying spamming botnets using Botlab. NSDI, p.291-306.

Kaemarungsi, K., Yoskamtorn, N., Jirawannakool, K., *et al.*, 2009. Botnet statistical analysis tool for limited resource computer emergency response team. IEEE 5th Int. Conf. on IT Security Incident Management and IT Forensics, p.27-40. [doi:10.1109/IMF.2009.13]

Kalt, C., 2000. Internet Relay Chat: Architecture. Available from http://tools.ietf.org/html/rfc2810 [Accessed on Oct. 20, 2013].

Kang, B.B., Chan-Tin, E., Lee, C.P., *et al.*, 2009. Towards complete node enumeration in a peer-to-peer botnet. Proc. 4th Int. Symp. on Information, Computer, and Communications Security, p.23-34. [doi:10.1145/1533057.1533064]

Kang, J., Zhang, J.Y., 2009. Application entropy theory to detect new peer-to-peer botnet with multi-chart CUSUM. IEEE 2nd Int. Symp. on Electronic Commerce and Security, p.470-474.

Karagiannis, T., Broido, A., Brownlee, N., *et al.*, 2003. File-sharing in the Internet: a characterization of P2P traffic in the backbone. Technical Report, University of California, Riverside, USA.

Karagiannis, T., Broido, A., Faloutsos, M., 2004. Transport layer identification of P2P traffic. Proc. 4th ACM SIGCOMM Conf. on Internet Measurement, p.121-134. [doi:10.1145/1028788.1028804]

Karagiannis, T., Papagiannaki, K., Faloutsos, M., 2005. BLINC: multilevel traffic classification in the dark. *ACM SIGCOMM Comput. Commun. Rev.*, **35**(4):229-240. [doi:10.1145/1090191.1080119]

Karasaridis, A., Rexroad, B., Hoeflin, D., 2007. Wide-scale botnet detection and characterization. Proc. first Conf. on 1st Workshop on Hot Topics in Understanding Botnets, p.1-8.

Kassner, M., 2003. The Top 10 Spam Botnets: New and Improved. Available from http://www.techrepublic.com/blog/10-things/the-top-10-spam-botnets-new-and-improved/ [Accessed on June 6, 2013].

Keizer, G., 2008. Top Botnets Control 1M Hijacked Computers. Available from http://www.computerworld.com/article/2536378/security0/top-botnets-control-1m-hijacked-computers.html [Accessed on Sept. 8, 2013].

Kespersky, 2011. How to Detect and Remove the Rootkit TDL4. Available from http://infoaleph.wordpress.com/2011/07/03/como-detectar-y-borrar-el-rootkit-tdl4-tdssalureon/ [Accessed on June 20, 2013].

Kugisaki, Y., Kasahara, Y., Hori, Y., *et al.*, 2007. Bot detection based on traffic analysis. IEEE Int. Conf. on Intelligent Pervasive Computing, p.303-306.

Lee, C.P., 2009. Framework for Botnet Emulation and Analysis. PhD Thesis, Georgia Institute of Technology.

Leonard, J., Xu, S., Sandhu, R., 2009. A first step towards characterizing stealthy botnets. IEEE Int. Conf. on Availability, Reliability and Security, p.106-113.

Li, C., Jiang, W., Zou, X., 2009. Botnet: survey and case study. IEEE 4th Int. Conf. on Innovative Computing, Information and Control, p.1184-1187.

Li, Z., Goyal, A., Chen, Y., *et al.*, 2009. Automating analysis of large-scale botnet probing events. Proc. 4th Int. Symp. on Information, Computer, and Communications Security, p.11-22.

Liao, W.H., Chang, C.C., 2010. Peer to peer botnet detection using data mining scheme. IEEE Int. Conf. on Internet Technology and Applications, p.1-4.

Liu, D., Li, Y., Hu, Y., *et al.*, 2010. A P2P-botnet detection model and algorithms based on network streams analysis. IEEE Int. Conf. on Future Information Technology and Management Engineering, p.55-58.

Liu, F., Li, Z., Nie, Q., 2009. A new method of P2P traffic identification based on support vector machine at the host level. IEEE Int. Conf. on Information Technology and Computer Science, p.579-582.

Liu, L., Chen, S., Yan, G., *et al.*, 2008. BotTracer: execution-based bot-like malware detection. *In*: Information Security. Springer Berlin Heidelberg, p.97-113. [doi:10.1007/978-3-540-85886-7_7]

Livadas, C., Walsh, R., Lapsley, D., *et al.*, 2006. Using machine learning technliques to identify botnet traffic. Proc. 31st IEEE Conf. on Local Computer Networks, p.967-974.

Lu, W., Ghorbani, A.A., 2008. Botnets detection based on IRC-community. IEEE Global Telecommunications Conf., p.1-5.

Lu, W., Tavallaee, M., Ghorbani, A., 2009a. Automatic discovery of botnet communities on large-scale communication networks. Proc. 4th Int. Symp. on Information, Computer, and Communications Security, p.1-10.

Lu, W., Tavallaee, M., Rammidi, G., *et al.*, 2009b. BotCop: an online botnet traffic classifier. 7th IEEE Annual Communication Networks and Services Research Conf., p.70-77.

Madhukar, A., Williamson, C., 2006. A longitudinal study of P2P traffic classification. 14th IEEE Int. Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, p.179-188. [doi:10.1109/MASCOTS.2006.6]

Mador, Z., 2012. M86 Security Threat Report for the Second Half of 2011 is Now Available. Available from http://labs.m86security.com/2012/02/m86-security-threat-report-for-the-second-half-of-2011-is-now-available/ [Accessed on June 20, 2013].

Mansmann, F., Fischer, F., Keim, D.A., *et al.*, 2009. Visual support for analyzing network traffic and intrusion detection events using TreeMap and graph representations. Proc. Symp. on Computer Human Interaction for the Management of Information Technology, p.3.

Marko, P., Vilhan, P., 2012. Efficient detection of malicious nodes based on DNS and statistical methods. IEEE 10th Int. Symp. on Applied Machine Intelligence and Informatics, p.227-230.

Marry, W., 2010. Pushdo Botnet. Available from http://msmvps.com/blogs/harrywaldron/archive/2010/02/02/pushdo-botnet-new-ddos-attacks-on-major-web-sites.aspx.

Marupally, P.R., Paruchuri, V., 2011. Comparative Analysis and Evaluation of Botnet Command and Control Models. 24th IEEE Int. Conf. on Advanced Information Networking and Applications, p.82-89.

Masud, M.M., Al-Khateeb, T., Khan, L., *et al.*, 2008. Flow-based identification of botnet traffic by mining multiple log files. IEEE 1st Int. Conf. on Distributed Framework and Applications, p.200-206.

McCarty, B., 2003. Botnets: big and bigger. *IEEE Secur. Priv.*, **1**(4):87-90. [doi:10.1109/MSECP.2003.1219079]

McMillan, R., 2009. Experts Bicker over Conficker Numbers. Available from http://news.techworld.com/security/114307/experts-bicker-over-conficker-numbers/ [Accessed on Oct. 14, 2013].

McMillan, R., 2010. Spanish Police Take Down Massive Mariposa Botnet. Available from http://www.pcworld.com/article/190634/article.html [Accessed on June 20, 2013].

Messmer, E., 2009. America's 10 Most Wanted Botnets. Available from http://www.networkworld.com/news/2009/072209-botnets.html [Accessed on June 20, 2013].

Miller, C., 2008. The Rustock Botnet Spams Again. SC Magazine, July 25.

Miller, C., 2009. Researchers Hijack Control of Torpig Botnet. Available from http://www.scmagazine.com/researchers-hijack-control-of-torpig-botnet/article/136207/ [Accessed on June 2, 2013].

Mills, E., 2009. Experts: Gumblar Attack Is Alive, Worse than Conficker. Available from http://news.cnet.com/8301-1009_3-10251779-83.html [Accessed on Oct. 2, 2013].

Mockapetris, P., 1987. Domain Names—Concepts and Facilities. Available from http://tools.ietf.org/html/rfc1034 [Accessed on Dec. 5, 2013].

Morrison, T., 2012. Spam Botnets: the Fall of Grum and the Rise of Festi. Available from http://www.spamhaus.org/news/article/685/ [Accessed on Dec. 12, 2013].

Moscaritolo, A., 2010. Kraken Botnet Re-emerges 318,000 Nodes Strong. Available from http://www.scmagazineus.com [Accessed on Dec. 14, 2013].

Mukosaka, S., Koike, H., 2007. Integrated visualization system for monitoring security in large-scale local area network. IEEE 6th Int. Asia-Pacific Symp. on Visualization, p.41-44.

Mulliner, C., Seifert, J.P., 2010. Rise of the iBots: owning a telco network. IEEE 5th Int. Conf. on Malicious and Unwanted Software, p.71-80.

Murugan, S., Kuppusamy, K., 2011. System and methodology for unknown malware attack. IET Int. Conf. on Sustainable Energy and Intelligent Systems, p.803-804.

Musil, S., 2012. More than 600,000 Macs Infected with Flashback Botnet. Available from http://www.cnet.com/news/more-than-600000-macs-infected-with-flashback-botnet/ [Accessed on Oct. 5, 2014].

Nagaraja, S., Mittal, P., Hong, C., *et al.*, 2010. BotGrep: finding P2P bots with structured graph analysis. USENIX Security Symp., p.95-110.

Nazario, J., 2009. Politically motivated denial of service attacks. The Virtual Battlefield: Perspectives on Cyber Warfare, p.163-181.

Oberheide, J., Karir, M., Mao, Z., 2007. Characterizing dark DNS behavior. *In*: Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, p.140-156. [doi:10.1007/978-3-540-73614-1_9]

Panda Security, 2013. Firewall: Prevent Unknown Connections Between the Network and the Internet. Available from http://www.pandasecurity.com/enterprise/solutions/security-appliances/firewall [Accessed on Sept. 9, 2013].

Paranoid, 2004. The Dangers of HTTPS. Available from http://www.wilderssecurity.com/threads/the-dangers-of-https.31087/ [Accessed on Oct. 5, 2013].

Paxton, N., Ahn, G.J., Chu, B., *et al.*, 2007. Towards practical framework for collecting and analyzing network-centric attacks. IEEE Int. Conf. on Information Reuse and Integration, p.73-78. [doi:10.1109/IRI.2007.4296600]

Perdisci, R., Corona, I., Dagon, D., *et al.*, 2009. Detecting malicious flux service networks through passive analysis of recursive dns traces. IEEE Annual Computer Security Applications Conf., p.311-320.

Pham, V.H., Dacier, M., 2011. Honeypot trace forensics: the observation viewpoint matters. *Fut. Gener. Comput. Syst.*, **27**(5):539-546. [doi:10.1016/j.future.2010.06.004]

Plohmann, D., Gerhards-Padilla, E., Leder, F., 2011. Botnets: Detection, Measurement, Disinfection & Defence. The European Network and Information Security Agency (ENISA).

Plohmann, D., Gerhards-Padilla, E., Leder, F., 2011. Botnets: 10 Tough Questions. Available from https://www.enisa.europa.eu/activities/Resilience-and-CIIP/critical-applications/botnets/botnets-10-tough-questions [Accessed on Dec. 20, 2013].

Podrezov, A., 2013. F-Secure, Threat Description: Backdoor: W32/Agobot. Available from http://www.f-secure.com/v-descs/agobot.shtml [Accessed on June 20, 2014].

Press, W., 2013. Wordpress website targeted by hackers.

Proffitt, B., 2012. BotClouds: How Botnets Now Offer Crime-as-a-Service. Available from http://readwrite.com/2012/11/15/botclouds-how-botnets-now-offer-crime-as-a-service#awesm=~opWmkZjKTKOJBu [Accessed on Dec. 4, 2013].

Provos, N., 2004. A virtual honeypot framework. USENIX Security Symp.

Puri, R., 2003. Bots & Botnet: an Overview. SANS Institute.

Qiao, Y., Yang, Y., He, J., *et al.*, 2012. Detecting parasite P2P botnet in eMule-like networks through quasi-periodicity recognition. Information Security and Cryptology-ICISC, p.127-139.

Raff, A., 2012. Ramnit Goes Social. Available from http://www.seculert.com/blog/2012/01/ramnit-goes-social.html [Accessed on Dec. 5, 2013].

Raghava, N.S., Sahgal, D., Chandna, S., 2012. Classification of botnet detection based on botnet architechture. IEEE Int. Conf. on Communication Systems and Network Technologies, p.569-572.

Ramachandran, A., Feamster, N., 2006. Understanding the network-level behavior of spammers. *ACM SIGCOMM Comput. Commun. Rev.*, **36**(4):291-302. [doi:10.1145/1151659.1159947]

Ramachandran, A., Feamster, N., Dagon, D., *et al.*, 2006. Revealing botnet membership using DNSBL counter-intelligence. Proc. 2nd USENIX Steps to Reducing Unwanted Traffic on the Internet, p.49-54.

Rieck, K., Schwenk, G., Limmer, T., *et al.*, 2010. Botzilla: detecting the phoning home of malicious software. Proc. ACM Symp. on Applied Computing, p.1978-1984.

Rodríguez-Gómez, R.A., Maciá-Fernández, G., García-Teodoro, P., 2013. Survey and taxonomy of botnet research through life-cycle. *ACM Comput. Surv.*, **45**(4): 1-33.

Rrushi, J., Mokhtari, E., Ghorbani, A.A., 2011. A statistical approach to botnet virulence estimation. Proc. 6th ACM Symp. on Information, Computer and Communications Security, p.508-512.

Ruiter, J., Warnier, M., 2011. Privacy regulations for cloud computing: compliance and implementation in theory and practice. *In*: Computers, Privacy and Data Protection: an Element of Choice. Springer, p.361-376. [doi:10.1007/978-94-007-0641-5_17]

Saha, B., Gairola, A., 2005. Botnet: an overview. CERT-In, White Paper, CIWP-2005-05, 240.

Sanchez, F., Duan, Z., Dong, Y., 2012. Blocking spam by separating end-user machines from legitimate mail server machines. *Secur. Commun. Networks*, p.1-9.

Schiller, C., Binkley, J., 2007. Spybot.

Schiller, C., Binkley, J., Harley, D., *et al.*, 2011. Botnets—the Killer Web APP. Syngress, Rockland.

Schmudlach, M., 2009. Calculating the Size of the Downadup Outbreak. Available from http://forums.cnet.com/7723-6132_102-325455/virus-spyware-alerts-january-16-2009/ [Accessed on Aug. 7, 2013].

Schwartz, M.J., 2012. Zeus Botnet Eurograbber Steals $47 Million. Available from http://www.informationweek.com/attacks/zeus-botnet-eurograbber-steals-$47-million/d/d-id/1107673 [Accessed on Nov. 6, 2013].

Schwartz, M.J., 2013. Microsoft, FBI Trumpet Citadel Botnet Takedowns. Available from http://www.informationweek.com/attacks/microsoft-fbi-trumpet-citadel-botnet-takedowns/d/d-id/1110261 [Accessed on Nov. 8, 2013].

Sevcenco, S., 2012. SdBot. Available from http://www.symantec.com/security_response/writeup.jspdocid=2002-051312-3628-99 [Accessed on Dec. 14, 2013].

Shahrestani, A., Feily, M., Ahmad, R., *et al.*, 2009. Architecture for applying data mining and visualization on network flow for botnet traffic detection. IEEE Int. Conf. on Computer Technology and Development, p.33-37.

Shin, Y.H., Im, E.G., 2009. A survey of botnet: consequences, defenses and challenges. Joint Workshop on Internet Security, p.1-11.

Silva, S.S., Silva, R.M., Pinto, R.C.G., *et al.*, 2013. Botnets: a survey. *Comput. Networks*, **57**(2):378-403. [doi:10.1016/j.comnet.2012.07.021]

Sousa, R., Rodrigues, N., Salvador, P., *et al.*, 2012. Analyzing the behavior of top spam botnets. IEEE Int. Conf. on Communications, p.6540-6544.

Spider, I.O., 2013. Discovered: Botnet Costing Display Advertisers over Six Million Dollars per Month. Available from http://www.spider.io/blog/2013/03/chameleon-botnet/ [Accessed on Dec. 14, 2013].

Stalmans, E., Irwin, B., 2011. A framework for DNS based detection and mitigation of malware infections on a network. IEEE Information Security South Africa, p.1-8.

Stefan, 2013. Sinkholing the Hlux/Kelihos Botnet—What Happened? Available from http://www.securelist.com/en/blog/208214147/Sinkholing_the_Hlux_Kelihos_botnet_what_happened [Accessed on Dec. 16, 2013].

Stephens, K., 2010. Malware Command and Control Overview. Technical Report. Available from http://www.nsci-va.org/whitepapers.htm [Accessed on Dec. 1, 2013].

Stewart, J., 2007. Pushdo – Analysis of a Modern Malware Distribution System. Available from http://www.secureworks.com [Accessed on Aug. 7, 2013].

Stewart, J., 2009. Spam Botnets to Watch in 2009. Dell SecureWorks. Available from http://www.secureworks.com/cyber-threat-intelligence/threats/botnets2009/ [Accessed on Nov. 5, 2013].

Stinson, E., Mitchell, J.C., 2007. Characterizing bots' remote control behavior. *In*: Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, p.89-108. [doi:10.1007/978-3-540-73614-1_6]

Stinson, E., Mitchell, J.C., 2008. Characterizing bots' remote control behavior. *In*: Botnet Detection. Springer, p.45-64. [doi:10.1007/978-0-387-68768-1_3]

Strayer, W.T., Walsh, R., Livadas, C., *et al.*, 2006. Detecting botnets with tight command and control. Proc. 31st IEEE Conf. on Local Computer Networks, p.195-202.

Strayer, W.T., Lapsely, D., Walsh, R., *et al.*, 2008. Botnet detection based on network behavior. *In*: Botnet Detection. Springer, p.1-24. [doi:10.1007/978-0-387-68768-1_1]

Stringhini, G., Holz, T., Stone-Gross, B., *et al.*, 2011. BOTMAGNIFIER: Locating Spambots on the Internet. USENIX Security Symp.

Symantic, 2010. Bagle. Available from http://www.messagelabs.com/mlireport/MLI_2010_04_Apr_FINAL_EN.pdf [Accessed on Apr. 7, 2014].

Systems, C., 2012. DNS Best Practices. Available from http://www.cisco.com/web/about/security/intelligence/dns-bcp.html [Accessed on Dec. 5, 2013].

Szongott, C., Henne, B., Smith, M., 2012. Evaluating the threat of epidemic mobile malware. IEEE 8th Int. Conf. on Wireless and Mobile Computing, Networking and

Communications, p.443-450.

Szymczyk, M., 2009. Detecting botnets in computer networks using multi-agent technology. IEEE 4th Int. Conf. on Dependability of Computer Systems, p.192-201.

Tartakovsky, A.G., Polunchenko, A.S., Sokolov, G., 2013. Efficient computer network anomaly detection by changepoint detection methods. *IEEE J. Sel. Topics Signal Process.*, 7(1):4-11. [doi:10.1109/JSTSP.2013.2241912]

The H Security, 2007. New Zealand Teenager Accused of Controlling Botnet of 1.3 Million Computers. Available from http://www.h-online.com/security/news/item/New-Zealand-teenager-accused-of-controlling-botnet-of-1-3-million-computers-734068.html

Thonnard, O., Dacier, M., 2011. A strategic analysis of spam botnets operations. Proc. 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conf., p.162-171. [doi:10.1145/2030376.2030395]

Tokhtabayev, A.G., Skormin, V.A., 2007. Non-stationary Markov models and anomaly propagation analysis in IDS. IEEE 3rd Int. Symp. on Information Assurance and Security, p.203-208. [doi:10.1109/IAS.2007.72]

Torre, J.D., 2013. Stealrat: an In-Depth Look at an Emerging Spambot Jessa. White Paper, Available from http://www.trendmicro.co.uk/media/wp/ stealrat-whitepaper-en.pdf

Traynor, P., Lin, M., Ongtang, M., *et al.*, 2009. On cellular botnets: measuring the impact of malicious devices on a cellular network core. Proc. 16th ACM Conf. on Computer and Communications Security, p.223-234.

Trend Micro, 2006. Taxonomy of Botnet Threats. Technical Report.

Trend Micro, 2013. Andrameda Botnet. Available from http://blog.trendmicro.com/trendlabs-security-intelligence/andromeda-botnet-gets-an-update/ [Accessed on Nov. 7, 2013].

Truhanov, A., 2010. Russian Botnet Wants to Kill the Competitor. Available from http://safe.cnews.ru/news/top/index.shtml2010/02/10/379202 (in Russian).

Tung, L., 2011. Android DreamDroid Two: Rise of Laced APPs. Available from http://www.itnews.com.au/News/259147/android-dreamdroid-two-rise-of-laced-apps.aspx [Accessed on May 5, 2013].

Vaarandi, R., 2013. Detecting anomalous network traffic in organizational private networks. IEEE Int. Multi-disciplinary Conf. on Cognitive Methods in Situation Awareness and Decision Support, p.2-9.

van Ruitenbeek, E., Sanders, W.H., 2008. Modeling peer-to-peer botnets. IEEE 5th Int. Conf. on Quantitative Evaluation of Systems, p.307-316.

Villamarín-Salomón, R., Brustoloni, J.C., 2008. Identifying botnets using anomaly detection techniques applied to DNS traffic. 5th IEEE Consumer Communications and Networking Conf., p.476-481.

Vishwanath, K.V., Vahdat, A., 2009. Swing: realistic and responsive network traffic generation. *IEEE/ACM Trans. Network.*, **17**(3):712-725. [doi:10.1109/TNET.2009.2020830]

Wang, B., Li, Z., Li, D., *et al.*, 2010. Modeling connections behavior for web-based bots detection. 2nd IEEE Int. Conf. on e-Business and Information System Security, p.1-4.

Wang, C., Li, T., Wang, H., 2009. Botnet detection based on analysis of mail flow. IEEE 2nd Int. Conf. on Biomedical Engineering and Informatics, p.1-4.

Wang, P., Sparks, S., Zou, C., 2007. An Advanced Hybrid Peer-to-Peer Botnet. Available from http://static.usenix.org/event/hotbots07/tech/full_papers/wang/wang_html [Accessed on June 6, 2013].

Wang, P., Sparks, S., Zou, C., 2010. An advanced hybrid peer-to-peer botnet. *IEEE Trans. Depend. Secur. Comput.*, **7**(2):113-127. [doi:10.1109/TDSC.2008.35]

Wang, W., Fang, B., Zhang, Z., *et al.*, 2009. A novel approach to detect IRC-based botnets. IEEE Int. Conf. on Networks Security, Wireless Communications and Trusted Computing, p.408-411.

Wang, X.R., 2003. Eggdrop. Available from http://www.symantec.com/security_response/writeup.jspdocid=2003-041013-5338-99 [Accessed on July 8, 2013].

Wang, Z., Wang, J., Huang, W., *et al.*, 2010. The detection of IRC botnet based on abnormal behavior. 2nd IEEE Int. Conf. on Multimedia and Information Technology, p.146-149.

Warner, G., 2010. Oleg Nikolaenko, Mega-D Botmaster to Stand Trial. Available from http://garwarner.blogspot.com/2010/12/oleg-nikolaenko-mega-d-botmaster-to.html

Weigle, M.C., Adurthi, P., Hernández-Campos, F., *et al.*, 2006. Tmix: a tool for generating realistic TCP application workloads in ns-2. *ACM SIGCOMM Comput. Commun. Rev.*, **36**(3):65-76. [doi:10.1145/1140086.1140094]

Welch, M.J., Cho, J., Olston, C., 2011. Search result diversity for informational queries. Proc. 20th Int. Conf. on World Wide Web, p.237-246. [doi:10.1145/1963405.1963441]

Wikipedia, 1998. NetBus. Available from http://en.wikipedia.org/wiki/NetBus [Accessed on Aug. 7, 2013].

Wikipedia, 2013a. Anomaly Detection. Available from http://en.wikipedia.org/wiki/Anomaly_detection [Accessed on Aug. 7, 2013].

Wikipedia, 2013b. Botnets. Available from http://en.wikipedia.org/wiki/Botnet [Accessed on Aug. 7, 2013].

Wikipedia, 2013c. Mariposa Botnet. Available from http://en.wikipedia.org/wiki/Mariposa_botnet [Accessed on Aug. 7, 2013].

Wills, C.E., Mikhailov, M., Shang, H., 2003. Inferring relative popularity of Internet applications by actively querying DNS caches. Proc. 3rd ACM SIGCOMM Conf. on Internet Measurement, p.78-90.

WordPress, 2008. Social VPN. Available from http://socialvpn.wordpress.com/ [Accessed on Dec. 25, 2013].

Wurzinger, P., Bilge, L., Holz, T., *et al.*, 2009. Automatically generating models for botnet detection. Computer Security ESORICS, p.232-249.

Xu, K., Yao, D., Ma, Q., *et al.*, 2011. Detecting infection onset with behavior-based policies. 5th IEEE Int. Conf. on Network and System Security, p.57-64.

Xu, Z., Chen, L., Gu, G., *et al.*, 2012. PeerPress: utilizing enemies' P2P strength against them. Proc. ACM Conf. on Computer and Communications Security, p.581-592.

Yen, T.F., Reiter, M.K., 2010. Are your hosts trading or plotting Telling P2P file-sharing and bots apart. IEEE 30th Int. Conf. on Distributed Computing Systems, p.241-252.

Ying, L., Yan, Z., Ou, Y.J., 2010. The design and implementation of host-based intrusion detection system. 3rd IEEE Int. Symp. on Intelligent Information Technology and Security Informatics, p.595-598. [doi:10.1109/IITSI.2010.127]

Yu, F., Xie, Y., Ke, Q., 2010. SBotMiner: large scale search bot detection. Proc. 3rd ACM Int. Conf. on Web Search and Data Mining, p.421-430. [doi:10.1145/1718487.1718540]

Yu, X., Dong, X., Yu, G., *et al.*, 2009. Online botnet detection by continuous similarity monitoring. IEEE Int. Symp. on Information Engineering and Electronic Commerce, p.145-149.

Yu, X., Dong, X., Yu, G., *et al.*, 2010. Online botnet detection based on incremental discrete Fourier transform. *J. Networks*, **5**(5):568-576. [doi:10.4304/jnw.5.5.568-576]

Zeidanloo, H.R., Manaf, A.A., 2009. Botnet command and control mechanisms. 2nd IEEE Int. Conf. on Computer and Electrical Engineering, p.564-568.

Zeidanloo, H.R., Shooshtari, M.J.Z., Amoli, P.V., *et al.*, 2010. A taxonomy of botnet detection techniques. 3rd IEEE Int. Conf. on Computer Science and Information Technology, p.158-162.

Zeng, Y., Yan, G., Eidenbenz, S., *et al.*, 2011. Measuring the effectiveness of infrastructure-level detection of large-scale botnets. IEEE 19th Int. Workshop on Quality of Service, p.1-9.

Zhang, J., Luo, X., Perdisci, R., *et al.*, 2011a. Boosting the scalability of botnet detection using adaptive traffic sampling. Proc. 6th ACM Symp. on Information, Computer and Communications Security, p.124-134.

Zhang, J., Perdisci, R., Lee, W., *et al.*, 2011b. Detecting stealthy P2P botnets using statistical traffic fingerprints. IEEE/IFIP 41st Int. Conf. on Dependable Systems & Networks, p.121-132.

Zhao, S., Lee, P.P., Lui, J., *et al.*, 2012. Cloud-based push-styled mobile botnets: a case study of exploiting the cloud to device messaging service. Proc. 28th Annual Computer Security Applications Conf., p.119-128.

Zhao, Y., Xie, Y., Yu, F., *et al.*, 2009. BotGraph: large scale spamming botnet detection. NSDI, **9**:321-334.

Zhou, L., Li, Z., Liu, B., 2006. P2P traffic identification by TCP flow analysis. IEEE Int. Workshop on Networking, Architecture, and Storages, p.2.

Zhu, Z., Lu, G., Chen, Y., *et al.*, 2008. Botnet research survey. 32nd Annual IEEE Int. Computer Software and Applications, p.967-972.

Zhuang, L., Dunagan, J., Simon, D.R., *et al.*, 2008. Characterizing botnets from email spam records. Proc. 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats LEET, Article 2**,** p.1-9.

Zou, C.C., Cunningham, R., 2006. Honeypot-aware advanced botnet construction and maintenance. IEEE Int. Conf. on Dependable Systems and Networks, p.199-208. [doi:10.1109/DSN.2006.38]