Survey and Taxonomy of Botnet Research through Life-Cycle

RAFAEL A. RODRÍGUEZ-GÓMEZ, GABRIEL MACIÁ-FERNÁNDEZ and
PEDRO GARCÍA-TEODORO, University of Granada

Of all current threats to cybersecurity, botnets are at the top of the list. In consequence, interest in this problem is increasing rapidly among the research community and the number of publications on the question has grown exponentially in recent years. This article proposes a taxonomy of botnet research and presents a survey of the field to provide a comprehensive overview of all these contributions. Furthermore, we hope to provide researchers with a clear perspective of the gaps that remain to be filled in our defenses against botnets. The taxonomy is based upon the botnet's life-cycle, defined as the sequence of stages a botnet needs to pass through in order to reach its goal.

This approach allows us to consider the problem of botnets from a global perspective, which constitutes a key difference from other taxonomies that have been proposed. Under this novel taxonomy, we conclude that all attempts to defeat botnets should be focused on one or more stages of this life-cycle. In fact, the sustained hindering of any of the stages makes it possible to thwart a botnet's progress and thus render it useless. We test the potential capabilities of our taxonomy by means of a survey of current botnet research, and find it genuinely useful in understanding the focus of the different contributions in this field.

Categories and Subject Descriptors: K.6.5 [**Security and Protection**]: Invasive Software; K.4.2 [**Social Issues**]: Abuse and Crime Involving Computers

General Terms: Security

Additional Key Words and Phrases: Attack, botnet, defense, detection, survey, taxonomy

ACM Reference Format:
Rodríguez-Gómez, R. A., Maciá-Fernández, G., and García-Teodoro, P. 2013. Survey and taxonomy of botnet research through life-cycle. ACM Comput. Surv. 45, 4, Article 45 (August 2013), 33 pages.
DOI: http://dx.doi.org/10.1145/2501654.2501659

1. INTRODUCTION

Botnets are one of the most serious current threats to cybersecurity. The term botnet is used to define a network of infected machines, termed bots, which are under the control of a human operator commonly known as the botmaster. Bots are used to carry out a wide variety of malicious and harmful actions against systems and services, including denial-of-service (DoS) attacks, spam distribution, phishing, and click fraud [Feily et al. 2009; Abu Rajab et al. 2006]. As an example of the impact of botnet attacks, the FBI recently disclosed that over $20 million in financial losses had been suffered in the USA alone. In one case, a victim confirmed a loss of nearly $20,000 caused by DoS attacks committed through botnets [FBI 2007].

This work has been partially supported by Spanish MICINN under project TEC2011-225790.
Authors' addresses: R. A. Rodríguez-Gómez (corresponding author), G. Maciá-Fernández, and P. García-Teodoro, CITIC-UGR, Department of Signal Theory, Telematics and Communications, University of Granada, C/Periodista Daniel Saucedo Aranda, s/n E-18071 Granada, Spain; email: rodgom@ugr.es.

DOI: http://dx.doi.org/10.1145/2501654.2501659

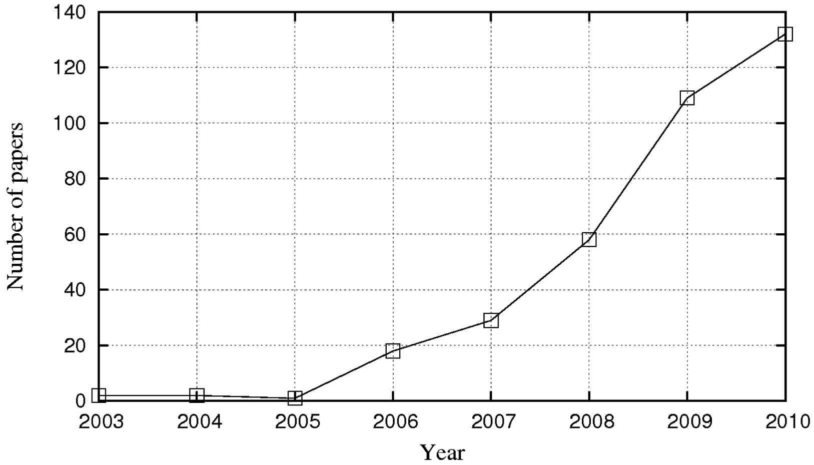

Fig. 1. Botnets research growth trend in last years.

Financial gains are usually the motive for the design and development of botnets by botmasters, who can reportedly make large sums by marketing their technical services. One example is that of Jeanson Ancheta, a 21-year-old hacker member of a group called the “Botmaster Underground”. He acquired more than $100,000 from different Internet advertising companies which paid him for introducing specially designed malicious adware code into more than 400,000 vulnerable PCs he had previously infected and taken over [Wilson 2007].

Highlighting the scope and thus the threat posed by botnets, Vinton Cerf, one of the “fathers of the Internet”, estimated that 100–150 million of the 600 million hosts on the Internet were part of a botnet [Weber 2007].

Between July 1st and December 31st, 2007, Symantec detected an average of 61,940 active bot-infected computers per day [Symantec 2008], which represents a 17% increase from the previous report. Symantec also observed around 5 million different bot-infected computers during this period, and 4,091 command and control (C&C) servers. In an additional study made between January 1st and March 31st, 2009, McAfee reported nearly 12 million new IP addresses operating as “zombies” [McAffe 2009], which is a significant increase over the levels in the Symantec report in 2007.

As a consequence of the impact of botnets, interest in this field is growing among the research community. As shown in Figure 1, the number of publications on botnets has grown exponentially in the last decade, from just a few in the early years of this century to more than a hundred in the last year.

Some taxonomies [Dagon et al. 2007; Zeidanloo et al. 2010] and surveys [Feily et al. 2009; Li et al. 2009a; Zhu et al. 2008; Bailey et al. 2009] on botnets have been proposed in recent years. However, these taxonomies mainly focus on different technical aspects of botnets, such as their architecture, communication protocols, or detection techniques, presenting a separate taxonomy for every one of these aspects. Accordingly, although these taxonomies provide a partial understanding of certain aspects of botnets, it is difficult to achieve a complete overview of the problem from them.

For this reason, there is a need to contribute a taxonomy that addresses the botnet problem from a global perspective. The present article aims to help put into perspective the huge amount of recent research in this field, highlighting the principal concerns and challenges presented.

In this context, we suggest it is useful to model a botnet from a product life-cycle perspective. As we extensively describe in this article, the botnet life-cycle begins with the conception of the botnet, and continues to its final objective of carrying out a certain attack. We believe this life-cycle is linear and composed of a set of different stages that are completed during the botnet's evolution.

Observing a botnet from a product life-cycle perspective allows us not only to understand the process of creation, development, integration, and use of a botnet, but also to organize the huge number of projects underway among the research community to defeat botnets. As a contribution to this goal, we present a survey on recent botnet research, following the proposed taxonomy.

More importantly, as explained shortly, this work has led us to the view that any defense technique or measure should be designed with the following idea in mind: "Hindering the normal execution of any of the stages in the botnet life-cycle renders the whole botnet useless."

Taking all these issues into consideration, the rest of the article is organized as follows. In Section 2 we present the principal concepts underlying botnet architecture. Section 3 introduces and describes our taxonomy, while Section 4 presents a selection of the most significant papers in the field of botnet research, organized according to the proposed taxonomy. Some of the main challenges currently being addressed in botnet research are then highlighted in Section 5, and finally, the main conclusions and contributions of this study are summarized in Section 6.

2. FUNDAMENTALS OF BOTNET ARCHITECTURE

In this section, we review the main concepts about botnets that will be used in the rest of the article. As previously stated, a botnet is a network of infected machines under the control of a human operator. Two main components can be found within a botnet: the bots and the botmaster. Infected machines are called *bots*, a term derived from the word robot, reflecting the fact that all bots follow the instructions given by the human operator, the *botmaster*, who controls the botnet and utilizes it to reach the ultimate goal, commonly that of carrying out a security attack against a victim. The botnet is controlled through the transmission of C&C messages among its members. To do so, *C&C channels* must be established. In some cases, bots connect to a *C&C server* in order to receive the messages sent by the botmaster. The existence of this server and the architecture of the C&C communications depends on the botnet's own architecture. The different known alternatives are described in Section 3.1. Apart from the cited main components of the botnet, that is, the botmaster and the bots, other roles appear during the life-cycle of botnets.

—*Developer.* This is the person or group of people who design and implement the botnet. The developer is not necessarily the same person(s) as the botmaster, as the design and implementation work could have been subcontracted. In this line, there exist several malware kits that provide all of the tools required to build up and administer a specific botnet. They are commonly named Do-it-Yourself (DIY) malware creation or generation kits. Several examples can be found: Zeus [Fortinet 2010], Twitter [Boyd 2010], Aldi [Danchev 2010], etc.

—*Client.* There exist two main types of clients of a botnet. Some clients rent botnet services from a botmaster. Examples of such services are spam distribution or Distributed Denial-of-Service (DDoS) execution to a certain server. On the other hand, some clients seek to become botmasters themselves, by gaining control of the botnet by an "illegal" commercial transaction. Subsequently, they use the botnet for their own purposes.

—*Victim.* This is the system, person, or network which constitutes the object of the attack executed. There are many different kinds of victims, depending on the main purpose of the botnet, that is, a user who receives spam, someone from whom confidential information is stolen, a company that loses several million dollars due to a DDoS attack, etc.

—*Passive participant*. This is the owner of a host which has been infected and therefore turned into a bot. This new bot becomes part of the botnet without the user's consent. This participation, even without the user's consent, can produce dramatic legal consequences, as in the case of Matthew Bandy, a 16-year-old boy, who could have received a 90-year prison sentence for (unwittingly) distributing child pornography [McElroy 2007].

Let us now focus on the specific particularities of botnets, emphasizing in the first place some of the main differences between them and other types of malware. These can be summarized as follows: (i) The botmaster can send orders to an infected host without directly controlling this machine; (ii) bots act in a coordinated manner and follow the botmaster's instructions. This ability to control a huge number of bots in a coordinated manner enables the botmaster to execute massive attacks, such as DDoS, click fraud, spam distribution, etc.

Additionally, botnets are designed in such a way that they attempt to hide all the preceding interactions, to hamper botnet detection processes. This is done using techniques like multihopping, ciphering, and binary obfuscation.

In the next section, these aspects are studied in greater detail and our proposal for a new taxonomy is presented.

3. BOTNET LIFE-CYCLE

This article explores the utilization of a botnet life-cycle as a basis for creating a taxonomy and performing a survey of the field. In fact, this concept has been proposed previously, and some papers cite the notion of a botnet life-cycle [Feily et al. 2009; Liu et al. 2009; Abu Rajab et al. 2006; Govil and Jivika 2007], but these studies merely illustrate some of the processes involved in the normal operation of a botnet. There is no comprehensive approach, either to the stages comprising the life-cycle or to the interaction between them. In other words, to the best of our knowledge, to-date there has been no in-depth study of these stages, how they should be characterized, and how they should be defined and distinguished.

Accordingly, we propose a detailed study of the botnet life-cycle, ranging from its conception to the achievement of the desired malicious purpose. The life-cycle proposed here is a linear sequence of stages, that is, the end of the life-cycle (the successful attack) is reached only after all the previous stages have been successfully carried out. This is an essential argument, as we seek to show that a failure in any of the intermediate stages of the life-cycle will thwart the botnet's aim.

In order to justify the aforesaid, we describe these stages in detail and which processes and roles are involved in each of them. Specifically, we define six stages in the botnet life-cycle, as shown in Figure 2. Although a more detailed specification of the different stages is presented later in the text, a brief description of them is advanced.

(1) *Botnet conception.* This is the first stage in any botnet life-cycle. The motivation for creating a botnet is an essential element that will directly affect its design and implementation. Diverse reasons might underlie a developer's decision to generate a botnet. Its main design characteristics are defined in this stage, and these are obviously influenced by the specific purpose intended for the botnet.

(2) *Botnet recruitment.* Once a botnet is conceived and created, there is a need for individual bots. Thus, a botmaster must recruit conventional hosts as members of the botnet.

(3) *Botnet interaction.* This stage involves two different processes. First, infected bots must be registered with the botnet in order to incorporate its dynamics and

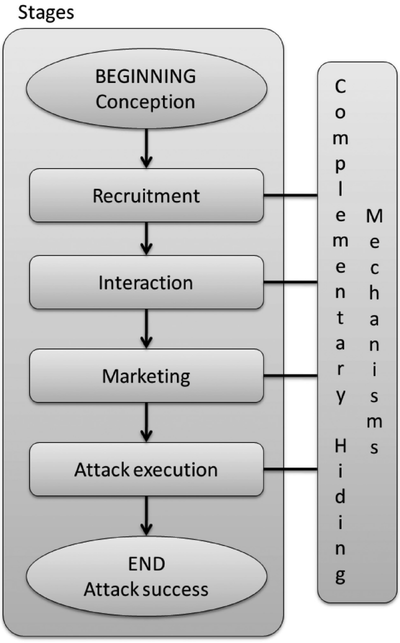

Fig. 2. Stages and complementary mechanisms of the proposed botnet life-cycle.

functioning. Second, there must be a communication framework supporting the operation and maintenance of the botnet, so that the botmaster may keep in contact with the different bots. These communications are mainly supported by a C&C channel. The information exchanged is constituted of orders (from the botmaster to the bots) and maintenance operations (code updating, membership accounting, etc.).

(4) *Botnet marketing*. Although the most common motivation for a developer to initiate the design and implementation of a botnet is to obtain monetary profit, there are many other possible reasons, including ego, specific cause, entrance to social groups, etc. Whatever the motivations, there is a marketing stage during which the botnet must be publicized; the developer needs to convey the advantages and capabilities of the botnet in a relevant forum in order to cede its use to clients and thus profit from it.

(5) *Attack execution*. In this stage, the botmaster orders the bots to perform an attack. As stated before, one of the main features of botnets is the huge number of bots recruited to carry out the malicious activities. Therefore, botnets are effective weapons for launching attacks that require a large number of hosts: DDoS, spam, click fraud, and phishing, among others.

(6) *Attack success*. The ultimate goal of a botnet is to successfully execute the attack for which it was designed.

Note that each stage of the life-cycle only represents the beginning of the execution of a specific process. Clearly, if the botnet reaches the *attack success* stage, the botmaster will try to apply it again for new attacks (conception stage); at the same time, new bots will continuously be recruited (recruitment stage) and controlled (interaction stage). For this reason, it should be understood that, according to this model, when a stage has been completed by a botnet, any process contained within or before that stage could subsequently be repeated.

As shown in Figure 2, there also exist various complementary mechanisms to the stages of the botnet life-cycle. These mechanisms are usually focused on attempting to conceal the botnet (the location of bots and the botmaster, the communication processes, etc.) from the security officer. Some examples of hiding mechanisms are multihopping, polymorphism, IP spoofing, and fast-flux networks, all described in Section 3.6.

As mentioned in the Introduction, much work is being done among the research community to defeat botnets and avoid their construction. These twin goals, however, require totally different strategies, as discussed shortly. However, at this point it is important to note that any single approach dedicated to defeating a botnet is really focused on preventing the execution of either a specific process in one of the preceding stages or that of a combination of processes in one or more stages. In consequence, it can be deduced that a measure that effectively prevents any of these stages from being executed is sufficient to thwart the botnet's success, for example, if a measure avoids the infection of bots it would be impossible to recruit enough soldiers to carry out the attack. In summary, if we can hinder the execution of just one stage in the botnet life, the goal of the botnet will be thwarted. For example, if the recruitment process is impeded by any defense mechanism, the botnet influence will be insignificant due to its limited scope. Obviously, most current botnets implement complementary hiding mechanisms that reduce the probability of being detected. Note that, in our life-cycle model, the hiding mechanisms implemented by botnets to hamper their detection are not considered as an additional stage. This decision was taken in coherence with our criterion for the definition of stages; preventing execution of hiding mechanisms does not necessarily imply the thwarting of the botnet goal, but only increases the probability that the botnet will be detected by a defense mechanism. In conclusion, we view the botnet life-cycle as a linear chain of stages in which a botmaster can be defeated by any one of the links (stages) being broken.

In the next sections, the different stages and processes of the botnet life-cycle are described in greater detail, considering the different design and implementation alternatives that developers or botmasters may adopt. This explanation also allows us to elaborate the taxonomy presented in Figure 3.

3.1. Conception

The first stage of the life-cycle is the *conception* of the botnet. It is important to understand the reasons underlying botnet creation and the usual architectures and designs employed.

The botnet conception stage can be divided into three phases or processes (see Figure 4): *motivation*, *design*, and *implementation*.

Motivation. A potential botmaster needs a good reason to create a botnet. The motivations of a botmaster have been classified as money, entertainment, ego, cause, entrance to social groups, and status: MEECES [Balas 2004]. However, the principal motivations are those related to financial gain. The Symantec Global Internet Security Threat Report, published in April 2010 [Symantec 2010], shows a detailed catalog of prices for botnet services. For example, the Zeus botnet kit costs around $700.

Whatever the reasons potential botmasters may have, the process following motivation is that of designing and implementing the desired botnet. Various aspects should be carefully considered during this process, especially those regarding bot infection and botnet communications (described in the next sections). However, a key decision regarding the design should be introduced at this point, namely the architecture of the botnet. This feature determines the operation of the subsequent elements, and thus of the botnet as a whole.

Design. Botnet architecture can be *centralized*, *distributed*, or *hybrid*. In a *centralized* scheme, bots contact the C&C server in order to receive information from the botmaster.

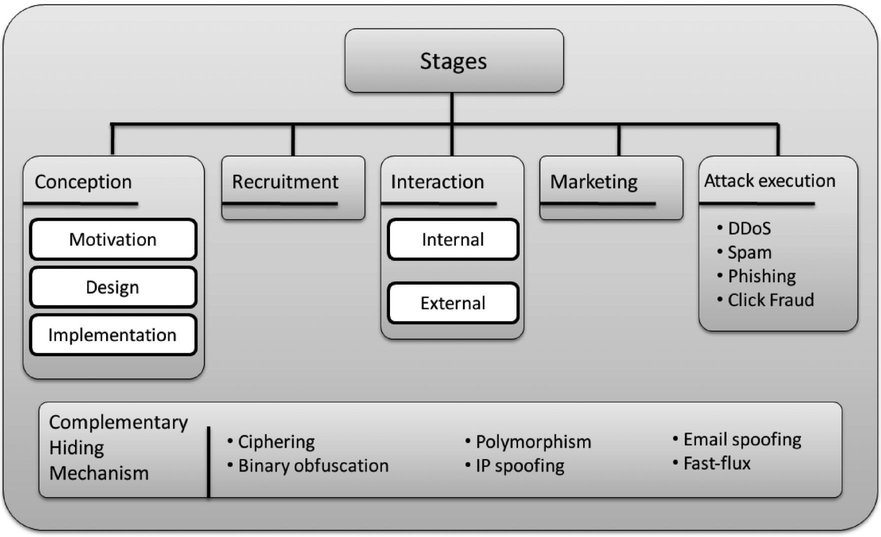

Fig. 3. Proposed taxonomy based on botnet life-cycle.

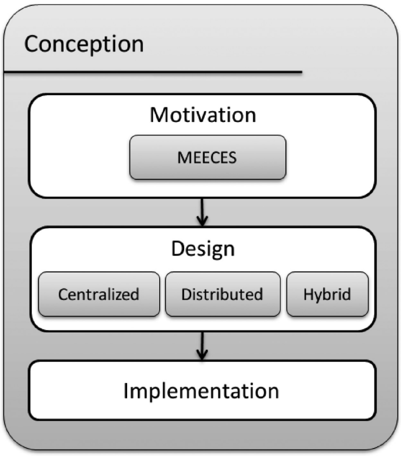

Fig. 4. Processes involved in the botnet conception stage.

In general, little time is spent in the transmission of a message from the botmaster to all the bots, and this represents one of the major advantages of this scheme. Its disadvantage is the fact that the C&C server constitutes a single point of failure. Thus, if the server shuts down, the complete network is dismantled. Examples of centralized botnets include Eggdrop [Pointer 1993], Gt-Bot and Agobot [Liu et al. 2009], and Bobax [Stewart 2004a].

In order to avoid this principal single point of failure limitation, a C&C architecture with fault tolerance is sometimes adopted. This is the situation of Conficker [Leder and Werner 2009], where the Domain Generation Algorithm (DGA) is implemented to determine in each case the central server that the bots are to be connected with.

In a *distributed* architecture, all the bots in the botnet act simultaneously as servers and clients. This philosophy avoids the existence of a single point of failure, and so this kind of botnet is more resistant than a centralized one. However, the time required for a message to reach all the nodes is much greater than in the centralized case. Many botnets present a distributed structure, including Spybot [Li et al. 2009a], Storm [Grizzard et al. 2007], Nugache [Stover et al. 2007], and Phatbot [Stewart 2004b].

Finally, *hybrid* botnets combine the advantages of the two previous architectures. In this case, there exist one or more distributed networks, each with one or more centralized servers. The disconnection of one of these servers implies, in the worst case, the fall of one of the distributed networks, allowing the rest of the botnet to continue its normal operation. Some examples of hybrid botnets are Torpig [Stone-Gross et al. 2009], Waledac [Sinclair et al. 2009], and a new design of botnet proposed by Wang et al. [2010a]. More recent and sophisticated hybrid botnets include Alureon/TDL4 [Rodionov and Matrosov 2011] and Zeus-P2P [abuse.ch abuse.ch abuse.ch abuse.ch 2011]. Alureon/TDL4 was the second most active botnet in the second quarter of 2010, according to a study by Microsoft. On the other hand, Zeus-P2P, also known as Murofet v2.0, is designed to steal personal information that is periodically sent to several centralized servers.

Implementation. Once the botnet has been conceptually conceived and designed, the last process involved in this stage concerns the implementation of the architecture. This task does not present special characteristics and can be performed following a traditional software development process.

3.2. Recruitment

The implemented botnet software must be deployed for operation in a real environment. For this purpose, bots must be recruited; indeed, the botmaster's aim is to recruit as many as possible. Note that this question is not unique to botnets, but is found in many cyberattack techniques. Recruitment, also known as *infection*, has been widely studied in the specialized literature.

In this respect, no special techniques are used in botnets, unlike those used by malware in general in its propagation. In order to increase these capabilities in bot software, it is usually designed to incorporate many existing exploits, and even new and unreported bugs (zero-day exploits) are explored by the software. Showing the scale of this problem, more than 16,000 vulnerabilities have been published in the last three years [NVD 2010]. As an example, the Stuxnet botnet [Chien 2010] was recently designed to exploit bugs in SCADA systems within nuclear plants. It is also remarkable that the Agobot botnet [Barford and Yegneswaran 2007] uses more than 10 exploits. For this reason, many subsequent botnet designs have been based on the Agobot code for recruitment, for example, Phatbot [Stewart 2004b].

According to Provos et al. [2009], in recent years, recruitment is mainly based on remotely exploiting servers' vulnerabilities, identified by scanning the Internet for vulnerable network services. Autonomously spreading a technique such as Conficker [Leder and Werner 2009] is an example of such a scanning attack. Other recruitment methods include sending out millions of email messages that contain something tempting to click on, or setting up Web sites with browser exploits that activate a drive-by-download [Provos et al. 2008] (a browser vulnerability allows a Trojan to be installed just by visiting this Web site) [Solomon and Evron 2006]. Currently, the propagation of malware through social networks is also growing rapidly as a recruitment mechanism [Faghani and Saidi 2009; Zetter 2009].

3.3. Interaction

This stage refers to all the interactions performed during the botnet operation, including the orders sent by the botmaster, the messages interchanged between bots,

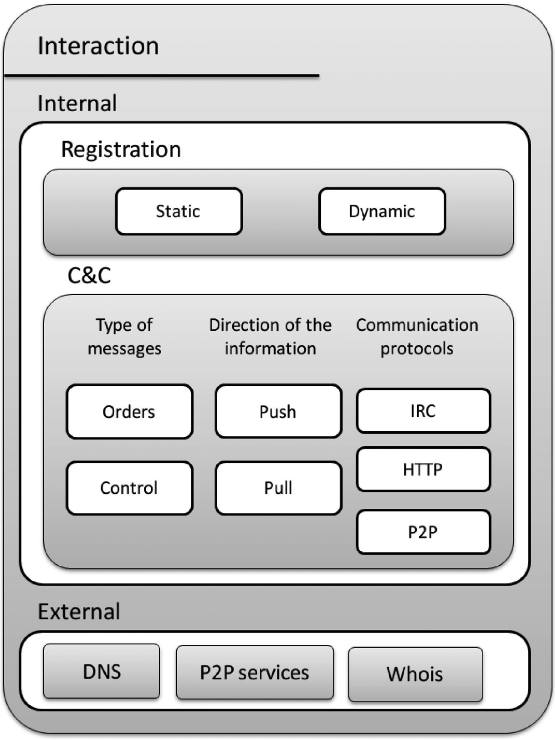

Fig. 5. Processes involved in the botnet interaction stage.

external communications from the botmaster to monitor botnet information, and also the communications of bots with external servers.

One of the main differences between botnets and other types of malware is the existence of communications making use of C&C channels. C&C messages are of particular significance, and many research papers on botnets are directly related to this question. Consequently, the botnet interaction stage is of major concern for the research community.

Figure 5 shows the processes involved in this third stage. These interaction processes can be classified as *internal* or *external*.

3.3.1. Internal Interactions. Internal interactions are those carried out among members of the botnet, that is, from the botmaster to the bots or vice versa, or only among bots. Here, we find two different processes: *registration* and *C&C communications*.

Registration process. Registration is the process through which a compromised host becomes an effective part of the botnet. This process is also required in other networks, like P2P, where the term *bootstrap* is used instead. There are two types of registration: *static* and *dynamic*.

In static registration, all the necessary information to become part of the botnet is hardcoded. Usually, the IP address of the C&C server is provided (with some type of obfuscation) in the code of the bots. GT-Bot, Agobot, and SDbot [Liu et al. 2009] are examples of this form of registration. In the case of distributed and hybrid botnets, the static registration is slightly more complex. A newly infected host uses a group of hardcoded IPs in order to request registration in the botnet. Waledac [Sinclair et al. 2009] is an example of this. Another example is Sinit [Stewart 2009], where every new bot tries several random IPs until it finds a member of the botnet, which then helps in the registration process.

In dynamic registration, bots have to explicitly request the necessary information to become part of the botnet, from a neutral third party or network. An example of this kind of registration is that used in Phatbot [Stewart 2004b], which utilizes Gnutella cache servers to download a list of peers that belong to the network. Once this is downloaded, peers are sampled until a member of the botnet is found. Then, the registration is made with this member.

C&C communications. The bulk of the interactions in the botnet occur after the registration process is completed. These interactions are the C&C communications, which can be grouped in accordance with three characteristics.

—*Type of messages*. As implied in the acronym, C&C messages can be classified as comprising either commands (orders) or control. Orders are generated by the botmaster telling bots to perform a certain action. On the other hand, control messages are mainly intended to compile information about the botnet, for instance, the number of active bots and their associated IP addresses (accountability).

—*Direction of the information*. C&C messages can be classified as *pull* or *push* [Gu et al. 2008b]. The bots periodically request information in pull C&C messages, but receive the information in a passive manner, without explicitly sending a previous request as is the case in the push C&C case.

—*Communication protocol*. Another important characteristic regarding C&C messages is the protocol(s) involved in the communications. The most common alternatives are the IRC, HTTP, and P2P protocols.

3.3.2. External Interactions. External interactions are those involving communications between a member of the botnet and a noncompromised system. These communications usually correspond to access to services commonly offered in the Internet. The following main external services are used by botnets.

—*DNS service*. In a centralized botnet, bots usually launch DNS queries to resolve the IP address of the C&C server. This is the case of the majority of IRC botnets, like Agobot, GT-Bot, or SDBot [Barford and Yegneswaran 2007], among others. There are also HTTP botnets, like Bobax [Stewart 2004a], which need DNS resolution.

—*Services in P2P networks.* A current trend in botnet design is the use of P2P networks as an intermediate layer to hide C&C communications. Instead of hardcoding the IP address of a C&C server, botmasters provide (in the bot code) only the name of a resource that should be downloaded from a P2P network. This resource contains a set of instructions from the botmaster, probably ordering an attack, or simply scheduling a new updating with the download of another resource from the P2P network. For example, Trojan.Peacomm [Grizzard et al. 2007] explores the Overnet network in search of keys generated by using a built-in algorithm. These keys correspond to files containing URLs from which bots will download an update file. Clearly, in these botnets, an important volume of traffic is generated in the P2P network by these interactions.

—*Botnet monitoring services.* By these means, a botmaster communicates with Internet services to obtain useful information about the botnet itself, for example, the availability of the domains used by the C&C server could be checked in a Whois server or DNS service [Ramachandran et al. 2006].

3.4. Botnet Marketing

At this stage, the botnet has been created and presents a good level of functionality. Now, the botmaster needs an incentive to use it. Although there are many possible reasons, such as entertainment, ego, status, etc., the most common is that of financial gain.

This expected profit is usually obtained by: (i) selling the botnet code or, more commonly, by (ii) renting out the services of the botnet. In both cases, an advertisement procedure is needed, through which the malicious user announces the capabilities and the services offered by the botnet.

Although botnet developers may sell the source code, it is more frequently distributed in the form of a DIY kit. As mentioned in Section 2, with such a kit even nonexpert users can generate malware binaries. One such kit is Turkojan 4, which can be obtained free of charge on the Internet [Turkojan 2012]. Similarly, the Zeus botnet kit is advertised in underground community forums for about $700, while the complete source code of NETTICK is offered for around $1200 [FBI 2010].

Another botnet marketing option is for services to be rented. Diverse services can be contracted; thus, a report from Namestnikov, Karspersky Lab [Namestnikov 2009] presents the following price list for renting the services of a botnet.

—*DDoS attacks*. This kind of attack costs from $50 to several thousand dollars per day, mainly depending on the size of the botnet, and thus the strength of the attack.
—*Email addresses*. Obtaining a list of around one million addresses costs from $20 to $100.
—*Sending spam email*. Sending spam email to a list of around a million addresses ranges from $150 to $200.
—*User accounts*. Obtaining the online service account information of users of hosts in which bots are located costs from $7 to $15 per account.
—*Fast-flux networks*. Cybercriminals, mostly phishers, pay botnet owners $1000 to $2000 per month for hosting fast-flux services.
—*Search engine spam*. Webmasters use search engine optimization techniques in order to improve their Web site ranking. This ranking could be improved in a fraudulent manner by using a botnet, at a cost of $300 per month.

3.5. Attack Execution

A botnet's ultimate goal is to execute an attack. The main feature of these attacks is the enormous number of attackers taking part. This does not mean that botnets cannot execute attacks conceived for a limited number of participants, but only that they are specifically designed for those requiring a large number.

The principal attacks launched by botnets are as follows.

—*Distributed Denial-of-Service (DDoS)*. DoS attacks are attempts to prevent the legitimate use of a service or simply reduce its availability. DDoS attacks are a particular case in which multiple attacking entities operate simultaneously (against one or more victims) to attain this goal [Mirkovic and Reiher 2004]. Botnets are perfectly suited for launching DDoS attacks. Defense measures based on filtering source IP addresses are bypassed, as the bots are very widely distributed, and usually do not share IP prefixes. Furthermore, they are so heterogeneous that they emulate the behavior of thousands of legitimate clients, making it extremely difficult to build defense schemes against a DDoS launched using botnets. Examples of botnets used for DDoS are Spybot and Agobot [Barford and Yegneswaran 2007].
—*Spamming*. A spam email contains certain information crafted to be delivered to a large number of recipients, whether they wish it or not [Cormack 2008]. The use of a botnet considerably increases the power to send a large number of spam emails in a few seconds. Moreover, they can be sent using SMTP servers associated with the users of the infected bots, and thus reception of the messages cannot be denied

merely by filtering out the sending server. Bobax is a botnet used for this purpose [Stewart 2004a].

—*Phishing*. This is a fraudulent activity defined as the creation of a replica of an existing Web page or other online resource to deceive a user into submitting personal, financial, or password data [van der Merwe et al. 2005]. These replicas involve a high development and maintenance cost and phishers put great effort into hiding them. For this purpose, they have recently been using a technique called fast-flux networks. This consists in acquiring a large number of proxies that redirect users' requests to the phishing server. These proxies change very frequently, by the use of DNS entries with low Time To Live (TTL) for the flux domain, thus making follow-up of the communications highly difficult. For a fast-flux network owner, it is important to have a high number of proxies available and it is also desirable for their locations to be heterogeneous (i.e., different networks). It is clear that bots in a botnet perfectly fit the role of proxies in a fast-flux network. Thus, botnets are used to implement this hiding mechanism. For example, the Storm botnet implements this mechanism to hide binary updating [Porras et al. 2007].

—*Data stealing*. This implies stealing sensitive information from users of the infected bots. This is done by using malware techniques like file inspection, keyloggers, cookie stealing, etc. As an example, the Zeus botnet [Stewart 2010] is able to steal cookies saved by Paypal and send them to the botmaster, among other actions.

—*Click fraud*. This consists in inducing, by deceit, users to click on online ads or to visit certain Web sites, and thus either increase third-party Web site revenues or exhaust an advertiser's budget [Wilbur and Zhu 2009]. The use of botnets makes it possible to simulate the behavior of millions of legitimate users, and is thus ideal for this kind of attack [Daswani and Stoppelman 2007].

3.6. Complementary Hiding Mechanisms

As shown in Figures 2 and 3, complementary hiding mechanisms are also considered part of the proposed botnet life-cycle, although they are not defined as a stage. These mechanisms are designed to hide the botnet and make it difficult to discover its components (bots, botmaster, C&C channels). Unfortunately, even if a hiding mechanism were uncovered, the botnet would not yet be defeated. For this reason, we consider that these mechanisms do not constitute a stage, but rather are complementary to them.

There are many possible hiding techniques. The following are the most widely used in botnets.

—*Multihopping*. The attacker connects to multiple proxies, specially located in various countries where access to the proxies' logs, before connecting to any bot, is hampered by legal restrictions. In consequence, tracking the final IP address is more difficult. This hiding scheme is typically used in the botnet interaction stage, such as in C&C communication with the bots. Multihopping might also appear in the marketing stage, for example, before publishing a message in a forum to announce the botnet.

—*Ciphering*. Regarding the interaction stage, in modern botnets, C&C communication channels are usually ciphered to prevent them from being analyzed. The development of viable techniques for detecting C&C channels becomes increasingly difficult when ciphering is used. SpamThru [Stewart 2006] and Zeus [Stewart 2010] use encrypted channels. Phatbot [Stewart 2004b] uses WASTE P2P encryption, a proprietary protocol developed for C&C communications in P2P networks.

—*Binary obfuscation*. Programmers use binary obfuscation techniques [Popov et al. 2007] to conceal the source code of a bot. This prevents reverse engineering and therefore the analysis of the bot behavior associated with the botnet implementation in the conception stage. The Conficker botnet [Leder and Werner 2009] makes

Table I. Complementary Hiding Mechanisms Present in Every Stage

	Multi-hop	Cipher	Binary Obf	Polymorph	Email spoof	IP spoof	Fast-flux
Conception			x				
Recruitment			x	x	x		x
Interaction	x	x					x
Marketing	x	x			x	x	x
Attack execution					x	x	x

intensive use of indirect calls and jumps and picks functions to pieces, which significantly complicates the analysis.

—*Polymorphism*. This consists in creating different versions of the source code of a program, which change while its functionality remains unaffected. This technique impedes the signature-based detection process used by most current antivirus tools. Polymorphism improves the botnet infection process during the recruitment stage. Phatbot [Stewart 2004b] and Zeus [Stewart 2010] botnets use this technique.

—*IP spoofing*. This consists in sending IP packets with a fake source address. It is widely used in DoS attacks with the aim of avoiding IP filters.

—*Email spoofing*. Similarly to IP spoofing, email spoofing consists in sending an email with a fake sender address (or other fields of the header). This is commonly used in phishing attacks, like that perpetrated by the Bobax botnet [Stewart 2004a].

—*Fast-flux network*. As explained in Section 3.5, this technique allows a final host to hide in the network. In a botnet, if a central C&C server exists, a fast flux created by a subset of the botnet could protect it.

Table I lists all these mechanisms and shows the stage in which they are used.

4. SURVEY OF BOTNET RESEARCH FROM A LIFE-CYCLE PERSPECTIVE

Our main intention in carrying out this survey of botnet research is to show how the huge number of research proposals related to botnet questions can be ordered and understood in accordance with our taxonomy. To do so, we now make a selection of the most significant papers directly related to the question of botnets and classify them into the different stages of the botnet life-cycle. With this work, we also seek to facilitate the design of defenses against botnets.

During the preparation of this taxonomy, we concluded that all attempts at defeating a botnet are really focused on one or more stages of its life-cycle. We now wish to demonstrate this by analyzing the most important contributions in this field.

The majority of papers on the subject of botnets are really focused on studying specific botnets or suggest defense mechanisms and algorithms against them. As we discussed in the corresponding stages, these two groups of studies are actually focused on the stages of botnet conception and interaction.

4.1. Botnet Conception

It seems clear that a good starting point to build a defense scheme would be to demotivate potential botnet developers from undertaking the task of designing a botnet. A first strategy to achieve this purpose is to implement specific legal measures aimed at demotivating botnet developers; see Section 3.1 in APEC [2008]. Another possible approach to hamper the implementation of botnets is to design networks which are resilient to their operation. However, to the best of our knowledge, this field is almost entirely unexplored.

In consequence, much research has focused on trying to understand botnet behavior and architecture, and how botnets are designed and implemented. This has been suggested as a prior step to discovering efficient detection mechanisms. Therefore, many papers on botnets are related to the botnet conception stage. We group them into three classes: botnet infiltration strategies, the study of botnets, and an understanding of new botnet designs.

4.1.1. Botnet Infiltration Strategies. In this group of contributions, researchers decide to become part of a botnet in order to study it from within. Generally, botnet infiltration is not an end purpose in itself but a means of studying a specific botnet. Although there are legal issues and technical problems involved in carrying out a real attack on a system, it is clear that this is a good way to learn several aspects of botnet behavior. The infiltration process is accomplished by becoming a normal bot, the most common approach, or a C&C server, thus taking control of part of the botnet.

Freiling et al. [2005] describe a general method for infiltrating as a bot in an IRC botnet: first, malware used by the botnet is obtained by using honeypots or Mwcollect. The latter is a program, similar to Honeyd [Provos 2004], developed by the latter authors to capture malware in nonnative environments. Then, the malware is analyzed to retrieve useful information which enables infiltration into the corresponding botnet. Cremonini and Riccardi [2009] present an open framework called Dorothy by which the activity of a botnet infiltration can be monitored. In a case study, they infiltrated and monitored a botnet called Siwa, collecting information about its functional structure, geographical distribution, communication mechanisms, command language, and operations.

The rendezvous mechanism of some botnets, like Conficker, Kraken, and Torpig, offers a unique opportunity to infiltrate as a C&C server. These botnets use domain flux, by which each bot periodically (and independently) generates a list of domains that are contacted. A bot then proceeds to contact them one after another. The first host that sends a reply identifying itself as a valid C&C server is considered genuine, until the next period of domain generation is started. Thus, to infiltrate these botnets, researchers have proposed methodologies to predict the set of domain names which are likely to be contacted by bots, and registered them to act as C&C servers. This is the case of Porras et al. [2009] with Conficker, Amini [2008] with Kraken, and Stone-Gross et al. [2011] with Torpig.

4.1.2. Study of Botnets. The study of botnets has been carried out by researchers at two levels: *bot level* and *network level*. The study at bot level consists in analyzing the actions performed by bots. The main aim of this study is to extract the flow diagram of a bot code, together with information characterizing every stage of the bot life-cycle, such as the communication protocol used, the attacks it can execute, the ciphering of its communications, etc. On the other hand, studies at a network level are fundamentally aimed at describing the general characteristics of a botnet: the number of online bots, the churn of the network (rate of members appearing or disappearing), the activity of a botnet over time, etc.

A detailed study of botnets is of enormous help in building defenses against them. For this reason, many papers have studied specific botnets, by means of different strategies. We classify these works into three main types: (i) tools to facilitate the study of botnets, (ii) the study of a particular botnet, and (iii) the study of certain characteristics of botnets in general.

Tools to facilitate the study of botnets. The use of reverse engineering to carry out a complete study of malware is an arduous task. To facilitate this, several proposals on automatic protocol reverse engineering have been developed. One of these, which has been directly applied to botnets, is Caballero et al. [2009], in which the authors propose the use of an automatic protocol reverse engineering process to reproduce the C&C messages in a botnet. This can subsequently be used for active botnet infiltration.

There are possibilities of analyzing botnets other than by reverse engineering a bot code. For example, Jackson et al. [2009] developed a system, called SLINGbot, that

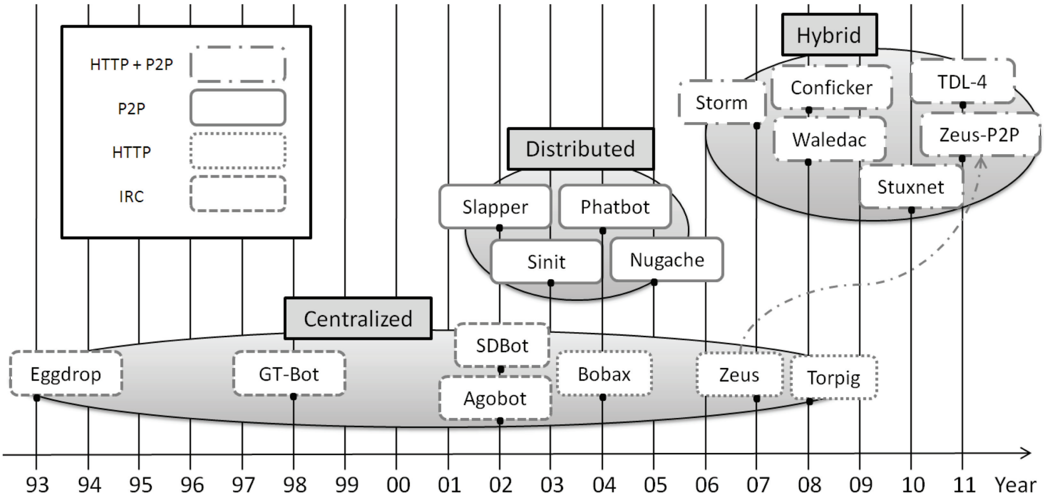

Fig. 6. Temporal evolution of botnets and their C&C protocols.

enables researchers to simulate current and potential future botnet traffic, characterize it, and thus derive effective defense techniques. Another example can be found in Cho et al. [2010], in which the authors propose a novel approach to infer protocol state machines in a realistic high-latency network setting, and apply it to the analysis of botnet C&C protocols.

Study of a particular botnet. Many research efforts have focused on studying a particular botnet. Taking a chronological approach, we highlight some of the most significant papers in this field (Figure 6). Eggdrop [Pointer 1993] is the first known IRC bot and was originally developed by Robey Pointer in 1993. Subsequently, different botnets with similar architectures appeared, such as GT-Bot [Barford and Yegneswaran 2007; Bacher et al. 2008] in 1998. This botnet was designed to launch an instance of the mIRC chat-client with a set of scripts and other binaries, while aHideWindow executable was used to make the mIRC instance invisible to the user. In 2002, Agobot [Barford and Yegneswaran 2007; Bacher et al. 2008] was the first bot to make use of a control protocol other than IRC. In the same year, SDBot [Barford and Yegneswaran 2007; Bacher et al. 2008] appeared. This was essentially a compact IRC implementation with its own IRC functions. In 2003, an enhanced version of SdBot appeared, called SpyBot [Barford and Yegneswaran 2007; Liu et al. 2009]. Its codebase is efficient, but it does not exhibit the modularity or breadth of capabilities of Agobot.

A natural evolution of the original IRC botnets was constituted by botnets based on the HTTP protocol. One of the major advantages of HTTP botnets is the ease of development and deployment. As an example, HTTP outbound traffic is always allowed and almost any compromised host will be able to communicate with an HTTP C&C server. Such botnets include Bobax, which appeared in 2004, ZeuS in 2007 (very active in 2010), and Torpig in 2008. Bobax bots are used as mail relay nodes, and they attempt to infect other machines by exploiting certain vulnerabilities as well as by using spam email [Stewart 2004a]. ZeuS and Torpig are designed to steal private information, mainly bank accounts, using phishing techniques [Stewart 2010; Stone-Gross et al. 2011]. Torpig uses domain flux to make it harder to take down the C&C server. Other HTTP botnets, such as Twetbot [Nazario 2009], use an external server (Twitter servers) to control their bots.

All the afore-described botnets are centralized botnets. As mentioned in Section 3.1, the main problem of centralized structures is the existence of unique points of failure located at C&C servers. Due to this feature, botnets have evolved from centralized network structures toward distributed paradigms. The first known botnet to utilize a P2P infrastructure was Slapper [Arce and Levy 2003] (2002). This is a variation of the Apache Scalper worm with the peculiarity of using a P2P protocol in its C&C communications via UDP through port number 2002. Later, in 2003, a botnet termed Sinit [Stewart 2009] appeared. This saves a list of known nodes and carries out a random bootstrap process, by which newly infected hosts try to connect to a random IP address until they find one that belongs to the botnet. Phatbot [Stewart 2004b] (2004) is a direct descendant of Agobot, and utilizes the Gnutella cache servers in order to conduct its bootstrap process. The messages sent in the P2P protocol are encrypted. In the Nugache botnet (2005) [Stover et al. 2007], a new bot joins the network using a hardcoded list of known peers and also communicates by means of an encrypted C&C channel.

Another recent issue regarding botnet architectures is the evolution to hybrid botnets, which constitute an intermediate solution between centralized and distributed structures. Storm Worm (also known as Peacomm) [Grizzard et al. 2007; Holz et al. 2008b; Stover et al. 2007] appeared in 2007 and was one of the most influential botnets to-date because of its massive distribution and the difficulty of rendering it unusable. Storm bots use Overnet, a decentralized P2P protocol, to find their HTTP C&C servers. Storm Worm was also among the first botnets to utilize fast-flux techniques. Its main purpose is the sending of spam mails. Two notable botnets appeared in 2008: Waledac and Conficker. Waledac [Calvet et al. 2009; il Jang et al. 2009; Sinclair et al. 2009] utilizes HTTP and P2P protocols for C&C and operates in a similar way to Storm. Conficker [Leder and Werner 2009; Shin and Gu 2010] also implements C&C communications on HTTP and P2P protocols and uses a domain flux to hide its C&C servers. In 2010, a hybrid botnet, Stuxnet [Chien 2010], appeared, capable of reprogramming industrial control systems (SCADA) by modifying code on programmable logic controllers. Its main purpose seems to be to launch attacks against nuclear power processing plants in Iran. The fourth version of TDL, TDL4 [Rodionov and Matrosov 2011], appeared in 2011 and it presents P2P- and centralized-based communications. A bootkit is installed in the infected system and loads the binary code before the operating system; thus, the AV detection is avoided. One difference between TDL4 and other hybrid botnets is the `kad.dll` library used to communicate with KAD, the distributed P2P network of the eMule client. This communication is carried out in a similar way to Storm, but adding a strong level of encryption. Finally, also in 2011, Zeus evolved to a hybrid architecture [abuse.ch abuse.ch abuse.ch abuse.ch 2011]. Zeus is now using a hardcoded "IP list" which contains IP addresses of other members in the P2P botnet. The botnet members communicate through an ad hoc P2P protocol to obtain the updates of the bot code and the configuration files. The bot then connects to the C&C domain listed in the configuration file using HTTP POST to send the stolen information.

Study of certain characteristics of botnets in general. Many recent studies have been made of certain general features of botnets, rather than focusing on a particular botnet. Some of these are related to the development of methods for crawling botnets and thus obtaining information about their characteristics. This is the case of Kanich et al. [2008b], in which the authors highlight a number of challenges that arise in using crawling to measure the size, topology, and dynamism of distributed botnets. These challenges include the existence of traffic generated by unrelated applications, address aliasing, and the presence of other active participants on the network such as poisoners. In Kang et al. [2009a], the authors present the Passive P2P Monitor (PPM), which can enumerate the infected hosts regardless of whether or not they are behind a firewall or NAT. Abu Rajab et al. [2006] constructed a multifaceted and distributed measurement infrastructure and, over a period of more than three months, used this infrastructure to track 192 unique IRC botnets. These authors revealed a number of behavioral and structural features of botnets, and subsequently, in Rajab et al. [2007], showed how various issues, including cloning, temporary migration, and hidden structures, can significantly increase the difficulty of accurately determining the size of a botnet.

4.1.3. Design of New Botnets. Finally, some papers have focused on defining possible alternatives in the design of new botnets. These studies allow the research community to develop new defenses that anticipate the creation of these new botnets. For instance, the authors of Wang et al. [2010a] describe a botnet with a hybrid structure. This botnet is composed of several supernodes that act as C&C servers to other distributed botnets. Jian et al. [2010] present a neighbors list selecting method based on a strongly connected graph, and analyze a supernode selecting mechanism based on AHP. The design of a P2P botnet with cryptography and a credit system is presented in Hund et al. [2008]. Starnberger et al. [2008] describe Overbot, a new P2P protocol based on existing legal P2P protocols related to Kademlia to carry out C&C communications. The messages between the bots are encoded using the `FIND_NODE` and `FIND_VALUE` functions, which allow a node to search for a specific value inside the DHT. Overbot uses the 160-bit hash values that are part of search queries to transmit a sequence number that can be used by the botmaster to publish commands. Nappa et al. [2010] present the design of a botnet based on Skype. Finally, Wang et al. [2010b] present some honeypot detection techniques to be used in both centralized botnets and peer-to-peer structured botnets.

4.2. Botnet Recruitment

Many proposals have been made by the research community to prevent the infection of hosts in the Internet, with particular attention being paid to the propagation of viruses and worms. When applied to the botnet problem, and in accordance with our taxonomy, all these techniques are located at the recruitment stage. The contributions presented in this case are divided into two classes: (i) studies of the recruitment process; and (ii) recruitment defense techniques.

4.2.1. Studies of the Recruitment Process. Many research projects have been aimed at studying the recruitment process in botnets, with the intention of determining useful characteristics that facilitate the development of defense techniques. These studies have revealed that the major contribution to malicious connection attempts can be directly attributed to botnet-related spreading activity. Indeed, this accounts for around 27% of all malicious connection attempts [Abu Rajab et al. 2006]. Li et al. [2009c] analyzed collections of malicious probing traffic in order to determine the significance of large-scale “botnet probes”. This analysis draws upon extensive honeynet data, the prevalence of different types of scanning, including properties such as trend, uniformity, or coordination among bots in the scanning process. It is important to highlight the work of Caballero et al. [2011], in which they perform a study of the Pay-Per-Install (PPI) market by infiltrating several PPI services. They find that 12 of the 20 most prevalent families of malware in 2010 make use of PPI services for recruitment purposes.

Several studies have proposed a propagation model. For example, Dagon et al. [2006] created a diurnal propagation model to examine how time and location affect botnet spreading dynamics. Another example is Li et al. [2009b], in which the authors build a propagation model for the Conficker botnet. This model takes into account the geography and connectivity among potential bots. Another approach is to study the propagation methods of specific botnets. For example, the authors in Harley et al. [2007] study the infection and propagation methods used by SDBot, Rbot, Agobot, Phatbot, Spybot, and Mytob.

Nowadays, drive-by-download infection is extremely common. As Provos et al. show in Provos et al. [2008], around 1.3% of all Google search queries produce at least one link pointing to a page that performs a drive-by-download attack. In Provos et al. [2007], the authors study Web-related malware by identifying the four main mechanisms used to inject malicious content into popular Web sites: Web server security, user-contributed content, advertising, and third-party widgets. Later, in Polychronakis et al. [2008] the life-cycle of Web-based malware is discussed. To capture the network profile of infected machines, the authors employ lightweight responders which provide fabricated responses for commonly used protocols such as SMTP, FTP, and IRC. In Shin et al. [2011], the authors study a large amount of infection data for Conficker, MegaD, and Srizbi, in search of similarities and differences between members of botnets with an auto-self-propagation method and botnets which need the help of external methods to be spread, such as PPI or drive-by-download.

4.2.2. Recruitment Defense Techniques. In the recruitment defense field, the bulk of contributions related to botnets are focused on detecting recruitment operations. Jordan et al. [2009] propose collecting and analyzing botnet growth patterns as they appear at the network level by intercepting the malware while it is transferred during infection. This has the tangible result of capturing malware in a pristine state. Gu et al. [2007] present BotHunter, a kind of network perimeter monitoring strategy, which focuses on recognizing the infection and coordination dialog that occurs during a successful bot infection. To counter drive-by-downloads, Egele et al. proposed in Egele et al. [2009] a detection mechanism integrated into the browser that relies on x86 instruction emulation to identify JavaScript string buffers containing shellcode.

4.3. Botnet Interaction

We believe that if a defense scheme could be developed to interrupt the processes involved in any one life-cycle stage, the botnet would be defeated. Obviously, this also applies to C&C communications, located at the interaction stage. Therefore, and because the existence of C&C communications is an inherent property of botnets, many researchers have devoted much time and effort in examining defense measures focused on the interaction stage. In fact, most of the papers on defense mechanisms against botnets are based on botnet interactions. To review this vast body of research with a certain degree of order, we classified them into four groups: (i) generic C&C detection; (ii) detection based on communication protocols; (iii) detection of communications with external services; and (iv) response to botnets.

4.3.1. Generic C&C Detection. The contributions cited in this group are those related to C&C detection in which no account is taken of the communication protocol used in the botnet.

Some proposals [AsSadhan et al. 2009; Pham and Dacier 2009; Gu et al. 2008a] are able to detect C&C traffic without specifying the communication protocol or network structure, or requiring any prior knowledge of botnets. In AsSadhan et al. [2009], the authors use periodograms to study the periodic behavior of these communications, and apply Walker's large sample test [Priestley 1982] to detect whether the traffic has a significant periodic component or not. The authors claim that if the traffic has a significant periodic component, it has been generated by a botnet. Pham and Dacier [2009] propose a method to identify and compile traces collected on low interaction honeypots by machines belonging to the same botnet(s). The Botminer detection framework [Gu et al. 2008a] clusters similar communication traffic and similar malicious traffic, and performs cross-cluster correlation to identify the hosts that share both similar communication patterns and similar malicious activity patterns. In Wurzinger et al. [2009], the authors present an automated mechanism to generate bot detection models by observing the actual behavior of bot instances in a controlled environment. These models are based on signs in the network traffic that a particular bot command is being sent, as reflected in the effects generated by the corresponding response.

In Gu et al. [2008b], the authors propose an approach called BotSniffer that uses network-based anomaly detection. In this case, the network structure is assumed centralized. BotSniffer identifies centralized botnet C&C channels in a local area network without any prior knowledge of signatures or C&C server addresses. This detection approach can identify both the C&C servers and the infected hosts in the network, and is based on the spatial-temporal correlation and the similarity of bots belonging to the same botnets. Yu et al. [2010b] propose transforming raw network traffic flows into multidimensional feature streams within a time window, after which they group feature streams with high similarities. Hosts classified into this group are then classified as potential bots. This approach can achieve efficient online botnet detection. Other authors [Chang and Daniels 2009] propose schemes to detect C&C channels of P2P botnets by observing node behavior profiles modeled by considering spatial and temporal correlations. Strayer et al. [2008] use bandwidth, packet timing, and burst duration to detect botnet C&C activity.

Some papers combine features extracted from one network with others of different origins or characteristics. For example, the authors in Zeng et al. [2010] present a framework that combines the network-level and the host-level information in order to determine whether the analyzed host is part of a botnet or not. A similar approach is taken in Efficient and Effective Bot Malware Detection (EFFORT) [Shin et al. 2012], where five modules are proposed to correlate information from host and network level.

Other authors [Ji et al. 2008] suggest combining of network characteristics extracted from several different locations (ISPs).

4.3.2. Detection Based on Communication Protocols. This category of contributions involves analyzing C&C communications taking into account the particularities of the protocols used. Most of these papers examine three kinds of communication protocols: IRC, HTTP, and P2P.

Many papers are based on detecting botnets that communicate by means of the IRC protocol. This is probably because, as mentioned before, it was the first protocol used in botnet communications. Mazzariello [2008] proposes a framework for detecting botnet activity that relies on a model of IRC user behavior. This framework detects and decodes IRC activity within raw network traffic and, by analyzing a set of descriptive parameters, builds up a classifier by which normal activity instances can be separated from botnet-related ones. In Wang et al. [2009b], the authors propose a novel algorithm called "channel distance", based on the similarity of nicknames in the same channel, to detect IRC-based botnets. This algorithm does not need any preanalysis of existing bots. In a similar way, Goebel and Holz [2007] implement Rishi, to detect IRC bots using techniques mainly based on passively monitoring network traffic for unusual or suspicious IRC nicknames, IRC servers, and uncommon server ports. To leverage these passive botnet C&C detection strategies, especially for small botnets with obfuscated C&C content and infrequent C&C interactions, the authors in Gu et al. [2009] present an IRC botnet detection method that uses active botnet probing techniques in a network middlebox. Binkley [2006] contributes an anomaly-based algorithm for detecting IRC-based botnet meshes. The algorithm is based on two tuples, one for determining the IRC mesh based on IRC channel names, and one to collect statistics on the number of SYN, FIN, and RESET packets observed.

Regarding HTTP botnet detection, Chen et al. [2010] develop a detection mechanism based on anomalous Web flow traffic over an administrative network domain. This is based on the idea that Web bots exhibit routine, regular Web connections which can be used to identify unusual Web flows within a network. Lee et al. [2008] also show the relations of HTTP clients with HTTP servers, and propose a method to identify malicious HTTP botnets by detecting traffic patterns with a certain periodicity.

Finally, some papers are related to the detection of P2P botnets. For example, Kang et al. propose [Kang et al. 2009b] a novel detection method using multi-chart CUSUM. Subsequently, Kang and Song [2010] describe a novel real-time detection model KCFM (Kalman filter and multi-chart CUSUM Fused Model) which uses the discrete Kalman filter to locate anomalous traffic, while the multi-chart CUSUM acts as the amplifier to make the abnormality more apparent. The aim of Masud et al. in Masud et al. [2008] is to detect P2P botnets using network traffic mining. A similar approach is taken by Liao and Chang [2010]. In the same line, Nagaraja et al. [2010] propose BotGrep, an algorithm that isolates peer-to-peer communication structures based on information about pairs of inter-communicating nodes (communication graph). The authors in Coskun et al. [2010] base their P2P bot detection method on identifying the network connections that involve known botnet members. They show that if bots select their peers randomly and independently, any given pair of P2P bots in a network communicate with at least one mutual peer outside the network with a surprisingly high probability. Zhang et al. [2011] present a detection system capable of detecting P2P botnets with no observable malicious activities, that is, stealthy P2P botnets. They first extract the hosts that expose a P2P behavior. Then, they derive statistical fingerprints of the P2P communications generated by these hosts. Finally, they distinguish between P2P normal hosts and P2P bots, extracting those with persistent connections and with a high overlap of the set of contacted IPs.

4.3.3. Detection of Communications with External Services. The contributions in this group are related to detecting the activity generated by a botnet while interacting with noncompromised servers. Several such proposals are based on detecting the communications of a botnet with DNS servers. In Villamarin-Salomon and Brustoloni [2008], the authors evaluate two approaches for identifying botnet C&C servers based on anomalous DDNS traffic. The first approach consists in looking for domain names whose query rates are abnormally high or temporally concentrated. The second approach searches for abnormally recurring DDNS replies indicating that the query is for a nonexistent name (NXDOMAIN). Yadav et al. propose in Yadav et al. [2010] an algorithm to detect the "domain fluxes" in DNS traffic by looking for patterns inherent to domain names that are generated algorithmically. To do this, they use the distribution of alphanumeric characters and the bigrams of all the domains mapped to the same group of IP addresses. Subsequently, in Yadav and Reddy [2011], the previous work is used in conjunction with an approach similar to Villamarin-Salomon and Brustoloni [2008]. Here, the detection method is based on the correlation between NXDOMAIN responses and the entropy of the domains belonging to the corresponding DNS queries. Choi et al. [2007] propose a botnet detection mechanism based on monitoring DNS traffic queries simultaneously sent by distributed bots. Ramachandran et al. [2006] perform counter-intelligence based on the insight that botmasters themselves perform DNSBL lookups to determine whether their spamming bots are blacklisted or not.

4.3.4. Response to Botnets. A commonly proposed response mechanism against botnets consists in filtering traffic coming from previously detected botnet members. Another approach is based on the detection of C&C servers, with the intention of bringing them down.

The bulk of these contributions are really focused on detection. However, some papers specifically focus on developing response mechanisms. For example, some proposals for responses against P2P botnets are based on the use of existent attacks in this type of network, relying on the use of a Sybil attack to thwart the botnet. Here, the attacker subverts the reputation system of a peer-to-peer network by creating a large number of pseudonymous entities, using them to gain a disproportionately large influence. Some contributions that explore this type of response are Holz et al. [2008b], Davis et al. [2008], and Davis et al. [2009].

First, in Holz et al. [2008b], Holz et al. presented a case study showing how to use Sybils to infiltrate the Storm botnet. These authors used an Overnet crawler which runs on a single machine. One of the main goals of this study was to determine the effect of pollution attacks (index poisoning) by posting polluted values associated with Storm search keys. The authors evaluated the effectiveness of the pollution attack by simultaneously polluting the value of a key used by Storm, and crawling the Storm network and searching for that key. Their experiment showed that by polluting the keys that Storm uses, they were able to disrupt the botnet communication.

The works of Davis et al. [2008, 2009] are complementary to that of Holz et al. An important difference with index poisoning attacks is that Sybil nodes must remain active and participate in the underlying P2P protocols, in order to remain in the peer list of regular bot nodes. However, they do not have to respond to the botmaster's commands or participate in illicit activities. In addition, these studies seek to discover how resilient such attacks are against botnet adjustments used as counter-countermeasures. Another work in this line is Ha et al. [2009], in which the authors simulate a P2P botnet based on Kademlia and evaluate the effectiveness of potential mitigation techniques: content poisoning, Sybil based, and Eclipse based.

4.4. Marketing

Defense schemes focusing on the marketing stage are a potentially effective point to undermine the effects of botnets. Taking legal measures is a possible approach. Obviously, if those who advertise botnets were subject to legal penalties, this could limit the use of botnets. Measures applied at the marketing stage should also focus on capturing and identifying botmasters' activities. Consider, for example, the case of Thomas James Frederick Smith in FBI [2010], one of the creators of *NETTICK* IRC botnet. He posted a public message on several forums in which he offered an executable program to control his botnet for $750. On 10 June 2010, he pled guilty to conspiracy to intentionally cause damage to a protected computer.

An example of a contribution in which defenders focus on the marketing stage can be found in Ramachandran et al. [2006]. These authors detect a botnet by inspecting the lookups made to the DNSBL, in which blacklisted domains are publicly available. These lookups are carried out by botmasters in order to assure the availability of their botnets, probably before renting their services.

The majority of the contributions from the research community are focused on understanding this new underground economy, as this is a crucial first step to designing appropriate defense schemes to thwart the marketing stage. In the following, we present some relevant contributions aimed at achieving a better understanding of this market, grouped into two sets: (i) advertisement analysis, and (ii) monetization analysis.

4.4.1. Advertisement Analysis. The first exploration of the underground economy of activities such as credit card fraud, identity theft, and spamming was carried out by Franklin et al. [2007]. They analyzed the messages interchanged in public Internet chat networks over 7 months and highlighted the importance of the shift from "hacking for fun" to "hacking for profit". Later, these markets evolved with a move toward Web forums and the sharing of wider information about malicious activities. In Zhuge et al. [2008], a detailed overview of this underground market and a model to describe it are presented. These authors crawl a black market forum to estimate the volume of criminal activity.

Some contributions analyze the behavior of participants in these underground forums. Thus, in Radianti [2010], the author highlights the interaction, rules, and social behavior of participants. In the same line, a system for automatically monitoring these channels and their participants is presented in Fallmann et al. [2010]. This system monitors both IRC and Web forums. Motoyama et al. empirically examine in Motoyama et al. [2011] the social networks formed in such forums and the mechanisms employed to manage trust, characterizing the social network makeup for six underground forums: BlackHatWorld, Carders, HackSector, HackE1ite, Freehack, and L33tCrew.

Other authors propose to focus on the defenses presented for other black markets. This is the case of Molnar et al. [2010], in which the authors propose "promises" and "puzzles" in the use of observational data for studying the Internet underground economy. These mechanisms have been encountered previously in drug policy research, and the paper highlights possible lessons to be learned from this different field.

4.4.2. Monetization Analysis. Several contributions focus on analyzing the question of monetization for a particular service, such as information stealing or spam. Holz et al. [2009] captured and analyzed a total of 33GB of keylogger data from more than 173,000 compromised machines. They focused on questions about the type and the amount of data stolen, and conclude that this type of cybercrime is a profitable business, allowing an attacker to earn hundreds of dollars per day. For example, they recovered more than 10,700 stolen online bank accounts, potentially worth several million dollars on the underground market.

Regarding the spam market, the authors in Kanich et al. [2008a] infiltrated an existing botnet's infrastructure. For nearly half a billion spam emails, they identified the number of user visits achieved to the advertised sites and the number of "sales" and "infections" produced. This work allows us to distinguish between claimed and true losses. This work was subsequently extended in Levchenko et al. [2011] where the authors analyzed large spam campaigns and examined the entirety of the value chain, but offered less precision regarding specific costs. In Kanich et al. [2011], the authors describe two inference techniques for probing within the business operations of spam-advertised enterprises: (i) estimating the number of orders received, and hence the revenue, via online store order numbering, and (ii) characterizing purchasing behavior through third-party image hosting data. Using this information, they were able to provide informed estimates of order volumes, product sales distribution, customer makeup, and total revenues for a range of spam-advertised programs.

4.5. Attack Execution

As described in Section 3.5, the most common attacks executed by botnets are DDoS, spam, phishing, and click fraud. Much research has been published on how to deal with these attacks, regarding prevention, detection, and response, but many open questions remain.

Although many contributions propose countermeasures against these attacks [Mirkovic et al. 2004; Douligeris and Mitrokotsa 2004], in this section we focus on those which are directly related to botnets.

The Botminer [Gu et al. 2008a] detection framework identifies as bots all the hosts that share similar malicious activity patterns and similar communication patterns.

The authors in Freiling et al. [2005] use traffic generated by DDoS attacks to detect botnet members. This is based on the observation that coordinated automated activity by many hosts needs a mechanism to remotely control them.

We also find in Yu et al. [2010a] the use of click fraud attack patterns to detect members of a botnet. The authors present SBotMiner, a system for automatically identifying stealthy, low-rate search bot traffic from query logs. This approach is able to capture groups of distributed, coordinated search bots. Duan et al. [2009] use spam activity to detect botnet members, and develop an effective spam zombie detection system named SPOT that monitors the outgoing messages of a network. The SPOT design is based on a powerful statistical tool called the Sequential Probability Ratio Test, by which false positive and false negative error rates are bounded. Similarly, the authors of Xie et al. [2008] characterize spamming botnets using spam payload and spam server traffic properties, and propose a spam signature generation framework called AutoRE to detect botnet-based spam emails and botnet membership. BotGraph [Zhao et al. 2009] is another spamming botnet detection capable of detecting stealthy botnet users in two steps. First, aggressive signups are detected, limiting the total number of accounts owned by a spammer. Second, a graph-based approach is used to detect the remaining stealthy bot accounts, based on the correlation between the login activities.

Al-Duwairi and Manimaran [2009] suggest the detection of DDoS using information provided by Web search engines (especially Googletrade). The main idea is that Googletrade can help distinguish human users from bot programs by directing users who want to access a Web site under attack to a group of nodes that will perform authentication in which users are required to solve a reverse Turing test to obtain access to the Web server. Another example of this approach is found in Collins et al. [2007]. Here, Collins et al. predict future hostile activity from past network activity. To do so, they define a network-based quality of uncleanliness, which is an indicator of how likely a network is to contain compromised hosts. They show that botnet activity predicts spamming and scanning.

4.6. Complementary Hiding Mechanisms

Malware normally uses complementary hiding mechanisms and much work has been done by the research community to detect and disrupt these mechanisms.

A significant type of hiding mechanism in the field of polymorphism and binary obfuscation is that of packers/cryptors. A packer [Yan et al. 2008] is a program to compress and encrypt executable files saved to disk, and to restore the original executable image when the packed file is loaded into memory. Packers are used to protect an application from being reverse engineered. This may be used by malware authors to obfuscate the structure of a malware file and thus avoid detection, as packing a single file using different packers results in syntactically different versions of the binary code. Although they can be used by legitimate software to minimize download times and storage space, or to protect copyrighted coding, they are commonly used in malware to disguise the contents of malicious files from malware scanners and to hamper the reverse engineering operations. As shown in some studies [Brosch and Morgenstern 2006], more than 92% of 735 malwares examined during 2006 were packed. In the same paper, the authors claim that the use of packers in malware can considerably reduce its detection rate (around 40% in some cases). Indeed, the packed malwares resulting from some packer applications are not detected at all, depending on the AV system, for example, UPX, Armadillo, Themida, VM-protect, Epack, Cexe, etc. The authors in Oberheide et al. [2008] also claim the questionable long-term effectiveness of traditional host-based antivirus methods due to the effectiveness of modern techniques such as polymorphism.

As mentioned in Section 3.6, the most common hiding mechanisms used by botnets are multihopping, ciphering, binary obfuscation, polymorphism, IP spoofing, email spoofing, and fast-flux networks. Here, we focus on fast-flux networks, because the use of this technique necessarily implies botnet support. This is due to the need for an enormous number of compromised hosts, that is, bot members. Assuming this, the detection of fast-flux participants is considered to be same problem as the detection of botnet members.

The first case study of fast-flux networks was presented in Honeynet Project [2007], in which several examples of fast-flux service networks were described. This paper also provided an overview of what fast-flux service networks are, how they operate, and how criminal communities leverage them. Subsequently, the first published system to detect fast-flux networks was FluXOR [Passerini et al. 2008]. The detection strategies of FluXOR rely entirely on the analysis of a set of features observable from the point of view of an ordinary user. The authors propose three types of features to distinguish between benign and malicious hostnames: domain name, availability of the network, and heterogeneity of the agents. Holz et al. [2008a] define a metric to distinguish fast-flux networks from legitimate content distribution networks. This metric is based on two restrictions for fast-flux networks: (i) the range of IP addresses of a fast flux is rather diverse, and (ii) there is no guaranteed uptime of the flux agent, as the controlled machines can go down anytime.

Nazario and Holz [2008] used data mining of live traffic to discover new fast-flux domains and then tracked those botnets with active measurements for several months. Caglayan et al. [2009a] present a behavioral analysis of fast-flux networks using their database compiled over a period of 9 months. Their results show that such networks share common life-cycle characteristics and form clusters based on size, growth, and type of malicious behavior.

Caglayan et al. in Caglayan et al. [2009b] present the first empirical study for detecting and classifying fast-flux service networks in real time. In Perdisci et al. [2009], the authors propose a novel, passive approach for detecting and tracking malicious flux service networks. Their detection system is based on the passive analysis of Recursive DNS (RDNS) traffic traces collected from multiple large networks. This method is capable of detecting malicious flux service networks in the wild, instead of extracting them from a spam mail or a domain blacklist. Bilge et al. [2011] present EXPOSURE, a system that employs large-scale, passive DNS analysis techniques to detect domains involved in malicious activities. They characterize different properties of DNS names and the ways that they are queried, extracting from the DNS traffic 15 features grouped into the following sets: time-based features, DNS answer-based features, TTL value-based features, and domain-name-based features. In the same line as EXPOSURE, Antonakakis et al. [2011] present Kopis, which monitors streams of DNS queries and responses from the upper DNS hierarchy to detect malware domain names. The features used by Kopis rely on the information accessible in the upper DNS hierarchy, thus enabling the detection of DNS malware-related domains even when no IP reputation information is present.

These contributions show that attackers are adopting fast-flux techniques, and that there is still much work to do. We suggest fast-flux networks should be studied in depth, because they constitute a promising way of detecting botnets.

5. FUTURE RESEARCH GUIDELINES

Despite the considerable number of proposals made and work carried out in the field of botnet detection and prevention, our study reveals that there are still certain aspects that should be investigated in greater depth. We now present the main challenges currently facing the research community in this field.

5.1. Design of Botnet-Resilient Protocols and Systems

Our survey of the contributions on botnets shows that the question of detection has been widely studied. However, preventing the appearance of botnets is almost totally unexplored.

A new trend in the design of botnets is the use of existent, legal P2P networks, which are named leeching botnets [Wang et al. 2009a]. This is the case of Storm [Grizzard et al.

2007], which uses the Overnet network to find and communicate with its bots. There exists also the example of Overbot [Starnberger et al. 2008], a new P2P protocol based on existent, legal P2P protocols related to Kademlia. Another example of a leeching botnet is Alureon/TDL4 [Rodionov and Matrosov 2011], one of the most active botnets in 2010, which communicates through KAD, the distributed P2P network of an eMule client.

In view of their heightened importance, we suggest that leeching botnets should be very carefully considered when future protocols and systems are designed. Up to now, engineers have taken into account the existence of DoS attacks and the problem of host infection, modifying their designs to take these two issues into consideration. However, it is also important to develop mechanisms to prevent the use of hidden C&C channels by botnet developers.

5.2. Cross-Stage Detection of Botnets

Despite the great effort made to detect botnets, this problem remains unsolved. Our examination of the botnet life-cycle suggests that any effective mechanism focused on hindering a particular stage constitutes a valid defense against botnets. On the other hand, another point of view can also be found in our approach to describe botnets. As in cross-layer design paradigms recently proposed for other technologies, a "cross-stage" approach could be followed for the detection of botnets. As an example, mechanisms for the detection of bot spreading (recruitment stage) could be combined with approaches to detect marketing actions by botnet developers (marketing stage).

BotMiner [Gu et al. 2008a] is the first approach to base detection on the assumption that bots continue communicating with some C&C servers/peers while performing malicious activities. These authors cross their information on the interaction and attack stages in order to carry out the detection process. However, as Zhang et al. [2011] claim, modern botnets tend to be stealthier in their malicious activities, making current detection approaches ineffective, including those in Gu et al. [2008a]. Therefore, new cross-stage approaches should be based not only on attack and interaction stages, but also on the different stages of the botnet life.

5.3. Defense against Botnets at the Marketing Stage

As mentioned in Section 4.4, the majority of the contributions discussing the marketing stage focus on monetization and advertisement analysis. These works aim at giving an idea of the importance of the botnet problem.

We believe that new defense schemes against botnets should be specifically based on the marketing stage. This is because nowadays botnets have shifted from "hacking for fun" to "hacking for profit" [Franklin et al. 2007] and therefore the marketing stage constitutes a visibility point through which botnets could be detected.

Thus, techniques should be developed to detect anomalous activity in a marketing environment: underground forums, social networks, etc. In a subsequent phase, these notifications should be correlated to direct the analysis toward attack vectors such as spam, DDoS, and click fraud.

5.4. Response Mechanisms against Botnets

The development of response mechanisms against the presence of botnets is also very scant. We believe the main reason for this is the lack of efficient detection mechanisms, which is a prior condition for a response to be made. Some approaches have been suggested in this field, for example, in Leder et al. [2009], where the authors propose joining the botnet and taking it down from within, like an autoimmune disease. Despite this and similar studies, the question has not been explored in depth and much more research is still required.

New response techniques could be based on existent prediction techniques and thus new botnets could be confronted from the very moment of their appearance. Some examples of such prediction techniques are Dagon et al. [2006], Collins et al. [2007], and Shin et al. [2011].

5.5. Moving from Size Estimation to Impact Estimation

Most botnet measurement studies concern the number of compromised bots within a particular botnet, that is, they are based on size estimation. As suggested in ENISA [2011], the research focus should shift from measuring the size of botnets towards analyzing the impact of specific botnets of particular significance to society.

It is essential to analyze malicious functionalities, especially the development of generic approaches for fast identification, and the handling of important aspects, including:

- C&C infrastructure and protocol;
- weaknesses and attack vectors in the malware and C&C infrastructure that can be used to allay it; and
- complexity of applied design concepts like cryptography, hiding techniques, and code obfuscation in the malware.

5.6. Exploring Infection based on Emerging Technologies

A common vector for the recruitment stage is based on exploiting software vulnerabilities. These vulnerabilities are particularly apparent new software or technologies. According to the technical report published by Juniper in February 2012 [Juniper 2012], in 2011 the number of android malware samples was 3,000% higher than in 2010. New botnets for Android include Geinimi and Droid Kungfu, among many others. Due to the extremely high and growing number of malwares based on emerging technologies, the research community must focus on this issue in order to reduce the effects of new infection techniques.

6. CONCLUSIONS

Because of the harmful effects of botnets and the considerable interest among the research community in this field, we propose a taxonomy of botnet research with a double aim: (i) to describe the botnet problem in global terms; and (ii) to provide a useful tool to clarify the wide variety of considerations on this question.

The taxonomy is based on the botnet's own life-cycle. This presents an interesting property: every stage of the life-cycle must be successfully completed if the botnet is to succeed. Therefore, interrupting the execution of just one stage in the botnet life-cycle renders the whole botnet useless.

We have reviewed current research work in this field, and show that all defense efforts are in fact focused on one or more of these stages. This review is presented here as a survey of the most relevant contributions in the field.

Our analysis of research contributions in this field has produced an overall picture and reveals the gaps needing to be filled in the near future. We also present our point of view about future research challenges, suggesting some main lines of approach.

REFERENCES

Abu-Rajab, M., Zarfoss, J., Monrose, F., and Terzis, A. 2006. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement (IMC'06).* ACM Press, New York, 41–52.

Abuse.Ch 2011. Zeus gets more sophisticated using P2P techniques. Tech. rep. http://www.abuse.ch/?p=3499.

Al-Duwairi, B. and Manimaran, G. 2009. Just-google: A search engine-based defense against botnet based ddos attacks. In *Proceedings of the IEEE International Conference on Communications (ICC'09).* 1–5.

AMINI, P. 2008. Kraken botnet infiltration. Tech. rep., DVLabs. http://dvlabs.tippingpoint.com/blog/2008/04/28/kraken-botnetinfiltration.

ANTONAKAKIS, M., PERDISCI, R., LEE, W., VASILOGLOU, I. N., AND DAGON, D. 2011. Detecting malware domains at the upper dns hierarchy. In *Proceedings of the 20th USENIX Conference on Security (SEC'11)*. USENIX Association, Berkeley, CA.

APEC. 2008. Guide on policy and technical approaches against botnet. Tech. rep., Telecommunications and Information Working Group, Asia-Pacific Economic Cooperation (APEC). http://publications.apec.org/publication-detail.php?pub_id=145.

ARCE, I. AND LEVY, E. 2003. An analysis of the slapper worm. *IEEE Secur. Privacy Mag. 1*, 1, 82–87.

ASSADHAN, B., MOURA, J. M. F., LAPSLEY, D., JONES, C., AND STRAYER, W. T. 2009. Detecting botnets using command and control traffic. In *Proceedings of the 8th IEEE International Symposium on Network Computing and Applications (NCA'09)*. 156–162.

BACHER, P., HOLZ, T., KOTTER, M., AND WICHERSKI, G. 2008. Know your enemy: Tracking botnets. Tech. rep., The Honeynet Project. October. http://www.honeynet.org/book/export/html/50.

BAILEY, M., COOKE, E., JAHANIAN, F., XU, Y., AND KARIR, M. 2009. A survey of botnet technology and defenses. In *Proceedings of the Cybersecurity Applications and Technology Conference for Homeland Security (CATCH'09)*. 299–304.

BALAS, E. 2004. *Know your Enemy: Learning about Security Threats* 2nd Ed. Addison Wesley.

BARFORD, P. AND YEGNESWARAN, V. 2007. An inside look at botnets. In *ARO-DHS Special Workshop on Malware Detection*, Advances in Information Security Series, vol. 27, Springer, 171–191.

BILGE, L., KIRDA, E., KRUEGEL, C., AND BALDUZZI, M. 2011. EXPOSURE: Finding malicious domains using passive dns analysis. In *Proceedings of the 18th Annual Network and Distributed System Security Symposium (NDSS'11)*.

BINKLEY, J. R. 2006. An algorithm for anomaly-based botnet detection. In *Proceedings of the USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI'06)*. 43–48.

BOYD, C. 2010. The diy twitter botnet creator. http://www.gfi.com/blog/the-diy-twitter-botnet-creator/.

BROSCH, T. AND MORGENSTERN, M. 2006. Runtime rackers: The hidden problem? Tech. rep., Black Hat. http://www.blackhat.com/presentations/bh-usa-06/BH-US-06-Morgenstern.pdf.

CABALLERO, J., GRIER, C., KREIBICH, C., AND PAXSON, V. 2011. Measuring pay-per-install: The commoditization of malware distribution. In *Proceedings of the 20th USENIX Conference on Security (SEC'11)*. USENIX Association, Berkeley, CA.

CABALLERO, J., POOSANKAM, P., KREIBICH, C., AND SONG, D. 2009. Dispatcher: Enabling active botnet infiltration using automatic protocol reverse-engineering. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09)*. ACM Press, New York, 621–634.

CAGLAYAN, A., TOOTHAKER, M., DRAPAEAU, D., BURKE, D., AND EATON, G. 2009a. Behavioral analysis of fast flux service networks. In *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW'09)*. 1–4.

CAGLAYAN, A., TOOTHAKER, M., DRAPEAU, D., BURKE, D., AND EATON, G. 2009b. Real-time detection of fast flux service networks. In *Proceedings of the Cybersecurity Applications and Technology Conference for Homeland Security (CATCH'09)*. 285–292.

CALVET, J., DAVIS, C., AND BUREAU, P.-M. 2009. Malware authors don't learn, and that's good! In *Proceedings of the 4th International Conference on Malicious and Unwanted Software (MALWARE'09)*. 88–97.

CHANG, S. AND DANIELS, T. E. 2009. P2p botnet detection using behavior clustering and statistical tests. In *Proceedings of the 2nd ACM Workshop on Security and Artificial Intelligence (AISec'09)*. 23–30.

CHEN, C.-M., OU, Y.-H., AND TSAI, Y.-C. 2010. Web botnet detection based on flow information. In *Proceedings of the International Computer Symposium (ICS'10)*. 381–384.

CHIEN, E. 2010. W32.stuxnet dossier. Tech. rep., Symantec. Septemeber. http://www.symantec.com/connect/blogs/w32stuxnet-dossier.

CHO, C. Y., BABIC, D., SHIN, E. C. R., AND SONG, D. 2010. Inference and analysis of formal models of botnet command and control protocols. In *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS'10)*. ACM Press, New York, 426–439.

CHOI, H., LEE, H., LEE, H., AND KIM, H. 2007. Botnet detection by monitoring group activities in dns traffic. In *Proceedings of the 7th IEEE International Conference on Computer and Information Technology (CIT'07)*. 715–720.

COLLINS, M. P., SHIMEALL, T. J., FABER, S., JANIES, J., WEAVER, R., SHON, M. D., AND KADANE, J. 2007. Using uncleanliness to predict future botnet addresses. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC'07)*. 93–104.

CORMACK, G. V. 2008. Email spam filtering: A systematic review. *Foundat. Trends Inf. Retriev. 1*, 4, 335–455.

Coskun, B., Dietrich, S., and Memon, N. 2010. Friends of an enemy: Identifying local members of peer-to-peer botnets using mutual contacts. In *Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC'10)*. ACM Press, New York, 131–140.

Cremonini, M. and Riccardi, M. 2009. The dorothy project: An open botnet analysis framework for automatic tracking and activity visualization. In *Proceedings of the European Conference on Computer Network Defense (EC2ND'09)*. 52–54.

Dagon, D., Gu, G., Lee, C., and Lee, W. 2007. A taxonomy of botnet structures. In *Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC'07)*. 325–339.

Dagon, D., Zou, C. C., and Lee, W. 2006. Modeling botnet propagation using time zones. In *Proceedings of the Network and Distributed System Security Symposium (NDSS'06)*.

Danchev, D. 2010. DIY botnet kit spotted in the wild. http://www.zdnet.com/blog/security/diy-botnet-kitspotted-in-the-wild/9440.

Daswani, N. and Stoppelman, M. 2007. The anatomy of clickbot.A. In *Proceedings of the 1st Conference on the 1st Workshop on Hot Topics in Understanding Botnets.* USENIX Association. 11.

Davis, C., Fernandez, J., and Neville, S. 2009. Optimising sybil attacks against P2P-based botnets. In *Proceedings of the 4th International Conference on Malicious and Unwanted Software (MALWARE'09)*. 78–87.

Davis, C., Fernandez, J., Neville, S., and Mchugh, J. 2008. Sybil attacks as a mitigation strategy against the storm botnet. In *Proceedings of the 3rd International Conference on Malicious and Unwanted Software (MALWARE'08)*. 32–40.

Douligeris, C. and Mitrokotsa, A. 2004. DDoS attacks and defense mechanisms: Classification and state-of-the-art. *Comput. Netw. 44*, 5, 643–666.

Duan, Z., Chen, P., Sanchez, F., Dong, Y., Stephenson, M., and Barker, J. 2009. Detecting spam zombies by monitoring outgoing messages. In *Proceedings of the 28th Conference on Computer Communications (INFOCOM'09)*. 1764–1772.

Egele, M., Wurzinger, P., Kruegel, C., and Kirda, E. 2009. Defending browsers against drive-by-downloads: Mitigating heap-spraying code injection attacks. In *Proceedings of the 6th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA'09).* Springer, 88–106.

Enisa. 2011. Botnets: Detection, measurement, disinfection and defence. Tech. rep., European Network and Information Security Agency (ENISA). http://www.enisa.europa.eu/activities/Resilience-and-CIIP/critical-applications/botnets/presentations-from-the-workshop-botnets-measurement-detection-disinfection-and-defence.

Faghani, M. and Saidi, H. 2009. Malware propagation in online social networks. In *Proceedings of the 4th International Conference on Malicious and Unwanted Software (MALWARE'09)*. 8–14.

Fallmann, H., Wondracek, G., and Platzer, C. 2010. Covertly probing underground economy marketplaces. In *Proceedings of the 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA'10)*. Springer, 101–110.

Fbi. 2007. Over one million potential victims of botnet cyber crime. Tech. rep., FBI Press Release. June. http://www.fbi.gov/news/pressrel/press-releases/over-1-million-potential-victims-of-botnet-cyber-crime.

Fbi. 2010. Another pleads guilty in botnet hacking conspiracy. Tech. rep., FBI Press Release. June. http://www.fbi.gov/dallas/press-releases/2010/dl061010.htm.

Feily, M., Shahrestani, A., and Ramadass, S. 2009. A survey of botnet and botnet detection. In *Proceedings of the 3rd International Conference on Emerging Security Information, Systems and Technologies (SECURWARE'09)*. 268–273.

Fortinet. 2010. Fortinet august threat landscape report shows return of ransomware and rise of "do-it-yourself" botnets. http://investor.fortinet.com/releasedetail.cfm?releaseid=504094.

Franklin, J., Perrig, A., Paxson, V., and Savage, S. 2007. An inquiry into the nature and causes of the wealth of internet miscreants. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07)*. ACM Press, New York, 375–388.

Freiling, F., Holz, T., and Wicherski, G. 2005. Botnet tracking: Exploring a root-cause methodology to prevent denial-of-service attacks. In *Proceedings of the 10th European Symposium on Research in Computer Security (ESORICS'05)*. 319–335.

Goebel, J. and Holz, T. 2007. Rishi: identify bot contaminated hosts by IRC nickname evaluation. In *Proceedings of the 1st Conference on the 1st Workshop on Hot Topics in Understanding Botnets.* USENIX Association, Berkeley, CA.

Govil, J. and Jivika, G. 2007. Criminology of botnets and their detection and defense methods. In *Proceedings of the IEEE International Conference on Electro/Information Technology*. 215–220.

GRIZZARD, J. B., SHARMA, V., NUNNERY, C., KANG, B. B., AND DAGON, D. 2007. Peer-to-peer botnets: Overview and case study. In *Proceedings of the 1st Conference on the 1st Workshop on Hot Topics in Understanding Botnets.* USENIX Association, Berkeley, CA, 1–8.

GU, G., PERDISCI, R., ZHANG, J., AND LEE, W. 2008a. BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *Proceedings of the 17th USENIX Security Symposium (Security'08).* 139–154.

GU, G., PORRAS, P., YEGNESWARAN, V., FONG, M., AND LEE, W. 2007. BotHunter: Detecting malware infection through IDS-driven dialog correlation. In *Proceedings of 16th USENIX Security Symposium.* 167–182.

GU, G., YEGNESWARAN, V., PORRAS, P., STOLL, J., AND LEE, W. 2009. Active botnet probing to identify obscure command and control channels. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC'09).* 241–253.

GU, G., ZHANG, J., AND LEE, W. 2008b. BotSniffer: Detecting botnet command and control channels in network traffic. In *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08).*

HA, D., YAN, G., EIDENBENZ, S., AND NGO, H. 2009. On the effectiveness of structural detection and defense against P2P-based botnets. In *Proceedings of the IEEE/IFIP International Conference on Dependable Systems Networks (DSN'09).* 297–306.

HARLEY, D., VIBERT, R. S., BECHTEL, K., BLANCHARD, M., DIEMER, H., LEE, A., MUTTIK, I., AND ZDRNJA, B. 2007. *AVIEN Malware Defense Guide for the Enterprise*. Elsevier.

HOLZ, T., ENGELBERTH, M., AND FREILING, F. 2009. Learning more about the underground economy: A case-study of keyloggers and dropzones. In *Proceedings of the 14th European Conference on Research in Computer Security (ESORICS'09).* Springer, 1–18.

HOLZ, T., GORECKI, C., FREILING, F., AND RIECK, K. 2008a. Measuring and detecting fast-flux service networks. In *Proceedings of the 15th Network and Distributed System Security Conference (NDSS'08).*

HOLZ, T., STEINER, M., DAHL, F., BIERSACK, E., AND FREILING, F. 2008b. Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm. In *Proceedings of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'08).* USENIX Association, 1–9.

HONEYNET PROJECT. 2007. Know your enemy: Fast-flux service networks. Tech. rep., The Honeynet Project. July. http://www.honeynet.org/book/export/html/130.

HUND, R., HAMANN, M., AND HOLZ, T. 2008. Towards next-generation botnets. In *Proceedings of the European Conference on Computer Network Defense*. 33–40.

IL JANG, D., KIM, M., CHUL JUNG, H., AND NOH, B.-N. 2009. Analysis of HTTP2P botnet: Case study waledac. In *Proceedings of the 9th Malaysia International Conference on Communications (MICC'09).* 409–412.

JACKSON, A., LAPSLEY, D., JONES, C., ZATKO, M., GOLUBITSKY, C., AND STRAYER, W. 2009. Slingbot: A system for live investigation of next generation botnets. In *Proceedings of the Cybersecurity Applications Technology Conference for Homeland Security (CATCH'09).* 313–318.

JI, S., IM, C., KIM, M., AND JEONG, H. 2008. Botnet detection and response architecture for offering secure internet services. In *Proceedings of the International Conference on Security Technology (SECTECH'08).* 101–104.

JIAN, G., YANG, Y., ZHENG, K., AND HU, Z. 2010. Research of an innovative P2P-based botnet. In *Proceedings of the International Conference on Machine Vision and Human-Machine Interface (MVHI'10).* 214–218.

JORDAN, C., CHANG, A., AND LUO, K. 2009. Network malware capture. In *Proceedings of the Cybersecurity Applications and Technology Conference for Homeland Security (CATCH'09).* 293–296.

JUNIPER. 2012. 2011 Mobile threats report. Tech. rep., Juniper Networks. February. http://www.juniper.net/us/en/local/pdf/additional-resources/jnpr-2011-mobile-threats-report.pdf.

KANG, B. B., CHAN-TIN, E., LEE, C. P., TYRA, J., KANG, H. J., NUNNERY, C., WADLER, Z., SINCLAIR, G., HOPPER, N., DAGON, D., AND KIM, Y. 2009a. Towards complete node enumeration in a peer-to-peer botnet. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (ASIACCS'09).* ACM Press, New York, 23–34.

KANG, J. AND SONG, Y.-Z. 2010. Detecting new decentralized botnet based on kalman filter and multichart cusum amplification. In *Proceedings of the 2nd International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC'10).* Vol. 1. 7–10.

KANG, J., ZHANG, J.-Y., LI, Q., AND LI, Z. 2009b. Detecting new P2P botnet with multi-chart cumsum. In *Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC'09).* Vol. 1. 688–691.

KANICH, C., KREIBICH, C., LEVCHENKO, K., ENRIGHT, B., VOELKER, G. M., PAXSON, V., AND SAVAGE, S. 2008a. Spamalytics: An empirical analysis of spam marketing conversion. In *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS'08).* ACM Press, New York, 3–14.

KANICH, C., LEVCHENKO, K., ENRIGHT, B., VOELKER, G. M., AND SAVAGE, S. 2008b. The heisenbot uncertainty problem: Challenges in separating bots from chaff. In *Proceedings of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'08)*. USENIX Association, 1–9.

KANICH, C., WEAVER, N., MCCOY, D., HALVORSON, T., KREIBICH, C., LEVCHENKO, K., PAXSON, V., VOELKER, G. M., AND SAVAGE, S. 2011. Show me the money: Characterizing spam-advertised revenue. In *Proceedings of the USENIX Security Symposium.*

LEDER, F. AND WERNER, T. 2009. Know your enemy: Containing conficker. Tech. rep., The Honeynet Project. April. http://www.honeynet.org/files/KYE-Conficker.pdf.

LEDER, F., WERNER, T., AND MARTINI, P. 2009. Proactive botnet countermeasures an offensive approach. In *Proceedings of the 1st Conference on Cyber Warfare (CCDECEO'09).*

LEE, J.-S., JEONG, H., PARK, J.-H., KIM, M., AND NOH, B.-N. 2008. The activity analysis of malicious httpbased botnets using degree of periodic repeatability. In *Proceedings of the International Conference on Security Technology (SECTECH'08).* 83–86.

LEVCHENKO, K., PITSILLIDIS, A., CHACHRA, N., ENRIGHT, B., FELEGYHE ZI, M., GRIER, C., HALVORSON, T., KANICH, C., KREIBICH, C., LIU, H., MCCOY, D., WEAVER, N., PAXSON, V., VOELKER, G. M., AND SAVAGE, S. 2011. Click trajectories: End-to-end analysis of the spam value chain. In *Proceedings of the IEEE Symposium on Security and Privacy.* 431–446.

LI, C., JIANG, W., AND ZOU, X. 2009a. Botnet: Survey and case study. In *Proceedings of the 4th International Conference on Innovative Computing, Information and Control (ICICIC'09).* 1184–1187.

LI, R., GAN, L., AND JIA, Y. 2009b. Propagation model for botnet based on conficker monitoring. In *Proceedings of the 2nd International Symposium on Information Science and Engineering (ISISE'09).* 185–190.

LI, Z., GOYAL, A., CHEN, Y., AND PAXSON, V. 2009c. Automating analysis of large-scale botnet probing events. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (ASIACCS'09).* 11–22.

LIAO, W.-H. AND CHANG, C.-C. 2010. Peer to peer botnet detection using data mining scheme. In *Proceedings of the International Conference on Internet Technology and Applications*. 1–4.

LIU, J., XIAO, Y., GHABOOSI, K., DENG, H., AND ZHANG, J. 2009. Botnet: Classification, attacks, detection, tracing, and preventive measures. *EURASIP J. Wirel. Comm. Netw. 2009*, 1.

MASUD, M. M., GAO, J., KHAN, L., HAN, J., AND THURAISINGHAM, B. 2008. Peer to peer botnet detection for cyber-security: A data mining approach. In *Proceedings of the 4th Annual Workshop on Cyber Security and Information Intelligence Research: Developing Strategies to Meet the Cyber Security and Information Intelligence Challenges Ahead (CSIIRW'08).* 1–2.

MAZZARIELLO, C. 2008. IRC traffic analysis for botnet detection. In *Proceedings of the 4th International Conference on Information Assurance and Security.* 318–323.

MCAFFE. 2009. Mcafee threats report: First quarter 2009. http://resources.mcafee.com/content/AvertReportQ109.

MCELROY, W. 2007. In child porn case, technology entraps the innocent. Tech. rep., Fox News.

MIRKOVIC, J., DIETRICH, S., DITTRICH, D., AND REIHER, P. 2004. *Internet Denial of Service. Attack and Defense Mechanisms*. Prentice Hall.

MIRKOVIC, J. AND REIHER, P. 2004. A taxonomy of ddos attack and ddos defense mechanisms. *SIGCOMM Comput. Comm. Rev. 34*, 2, 39–53.

MOLNAR, D., EGELMAN, S., AND CHRISTIN, N. 2010. This is your data on drugs: Lessons computer security can learn from the drug war. In *Proceedings of the Workshop on New Security Paradigms (NSPW'10).* ACM Press, New York, 143–149.

MOTOYAMA, M., MCCOY, D., LEVCHENKO, K., SAVAGE, S., AND VOELKER, G. M. 2011. An analysis of underground forums. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC'11).* ACM Press, New York, 71–80.

NAGARAJA, S., MITTAL, P., HONG, C.-Y., CAESAR, M., AND BORISOV, N. 2010. BotGrep: Finding P2P bots with structured graph analysis. In *Proceedings of the 19th USENIX Conference on Security*. USENIX Association, Berkeley, CA, 95–110.

NAMESTNIKOV, Y. 2009. The economics of botnets. Tech. rep., Securelist. July. http://www.securelist.com/en/downloads/pdf/ynam_botnets_0907_en.pdf.

NAPPA, A., FATTORI, A., BALDUZZI, M., DELLAMICO, M., AND CAVALLARO, L. 2010. Take a deep breath: A stealthy, resilient and cost-effective botnet using skype. In *Proceedings of the 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA'10).* Springer, 81–100.

NAZARIO, J. 2009. Twitter-based botnet command channel. Tech. rep., Arbor SERT. August. http://ddos.arbornetworks.com/2009/08/twitter-based-botnet-command-channel/.

NAZARIO, J. AND HOLZ, T. 2008. As the net churns: Fast-flux botnet observations. In *Proceedings of the 3rd International Conference on Malicious and Unwanted Software (MALWARE'08)*. 24–31.

NVD. 2010. Vulnerabilities in the last three years. Tech. rep., National Vulnerability Database. http://nvd.nist.gov/.

OBERHEIDE, J., COOKE, E., AND JAHANIAN, F. 2008. CloudAV: N-version antivirus in the network cloud. In *Proceedings of the 17th Conference on Security Symposium (SS'08)*. USENIX Association, Berkeley, CA, 91–106.

PASSERINI, E., PALEARI, R., MARTIGNONI, L., AND BRUSCHI, D. 2008. FluXOR: Detecting and monitoring fast-flux service networks. In *Proceedings of the 5th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA'08)*. 186–206.

PERDISCI, R., CORONA, I., DAGON, D., AND LEE, W. 2009. Detecting malicious flux service networks through passive analysis of recursive DNS traces. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC'09)*. 311–320.

PHAM, V.-H. AND DACIER, M. 2009. Honeypot traces forensics: The observation viewpoint matters. In *Proceedings of the 3rd International Conference on Network and System Security (NSS'09)*. 365– 372.

POINTER, R. 1993. Home page of eggdrop botnet. http://s23.org/wiki/Eggdrop.

POLYCHRONAKIS, M., MAVROMMATIS, P., AND PROVOS, N. 2008. Ghost turns zombie: Exploring the life cycle of web-based malware. In *Proceedings of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats*. USENIX Association, Berkeley, CA, 11:1–11:8.

POPOV, I. V., DEBRAY, S. K., AND ANDREWS, G. R. 2007. Binary obfuscation using signals. In *Proceedings of the 16th USENIX Security Symposium*. USENIX Association, 275–290.

PORRAS, P., SAIDI, H., AND YEGNESWARAN, V. 2009. A foray into Conficker's logic and rendezvous points. In *Proceedings of the 2nd USENIX Conference on Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More (LEET'09)*. USENIX Association, 1–9.

PORRAS, P., SAIDI, H., AND YEGNESWARAN, V. 2007. A multiperspective analysis of the storm (peacomm) worm. Tech. rep., Cyber-ta project page. http://www.cyber-ta.org/pubs/StormWorm/SRITechnical-Report-10-01-Storm-Analysis.pdf.

PRIESTLEY, M. B. 1982. *Spectral Analysis and Time Series*. Academic Press.

PROVOS, N. 2004. A virtual honeypot framework. In *Proceedings of the 13th USENIX Security Symposium (SSYM'04)*. Vol. 13. USENIX Association, 1–14.

PROVOS, N., MAVROMMATIS, P., RAJAB, M. A., AND MONROSE, F. 2008. All your IFRAMES point to us. In *Proceedings of the 17th Conference on Security Symposium*. USENIX Association, Berkeley, CA, 1–15.

PROVOS, N., MCNAMEE, D., MAVROMMATIS, P., WANG, K., AND MODADUGU, N. 2007. The ghost in the browser analysis of web-based malware. In *Proceedings of the 1st Conference on the 1st Workshop on Hot Topics in Understanding Botnets*. USENIX Association, Berkeley, CA, USA.

PROVOS, N., RAJAB, M. A., AND MAVROMMATIS, P. 2009. Cybercrime 2.0: When the cloud turns dark. *Comm. ACM 52*, 42–47.

RADIANTI, J. 2010. A study of a social behavior inside the online black markets. In *Proceedings of the 4th International Conference on Emerging Security Information Systems and Technologies (SECURWARE'10)*. 189–194.

RAJAB, M. A., ZARFOSS, J., MONROSE, F., AND TERZIS, A. 2007. My botnet is bigger than yours (maybe, better than yours): Why size estimates remain challenging. In *Proceedings of the 1st Conference on the 1st Workshop on Hot Topics in Understanding Botnets*. USENIX Association.

RAMACHANDRAN, A., FEAMSTER, N., AND DAGON, D. 2006. Revealing botnet membership using DNSBL counter-intelligence. In *Proceedings of the 2nd Conference on Steps to Reducing Unwanted Traffic on the Internet.* Vol. 2. USENIX Association, 49–54.

RODIONOV, E. AND MATROSOV, A. 2011. The evolution of tdl: Conquering x64. Tech. rep.,ESET. June. http://go.eset.com/us/resources/white-papers/The_Evolution_of_TDL.pdf.

SHIN, S. AND GU, G. 2010. Conficker and beyond: A large-scale empirical study. In *Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC'10)*. ACM Press, New York, 151–160.

SHIN, S., LIN, R., AND GU, G. 2011. Cross-analysis of botnet victims: New insights and implications. In *Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection (RAID'11)*.

SHIN, S., XU, Z., AND GU, G. 2012. EFFORT: Efficient and effective bot malware detection. In *Proceedings of the 31st Annual IEEE Conference on Computer Communications (INFOCOM'12)*.

SINCLAIR, G., NUNNERY, C., AND KANG, B.-H. 2009. The waledac protocol: The how and why. In *Proceedings of the 4th International Conference on Malicious and Unwanted Software (MALWARE'09)*. 69–77.

SOLOMON, A. AND EVRON, G. 2006. The world of botnets. *Virus Bull*. 10–12. http://www.beyondsecurity.com/whitepapers/SolomonEvronSept06.pdf.

STARNBERGER, G., KRUEGEL, C., AND KIRDA, E. 2008. Overbot: A botnet protocol based on kademlia. In *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks (SecureComm'08).* 1–9.

STEWART, J. 2004a. Bobax trojan analysis. Tech. rep., SecureWorks. http://www.secureworks.com/research/threats/bobax/.

STEWART, J. 2004b. Phatbot trojan analysis. Tech. rep., SecureWorks. http:// www.secureworks.com/research/threats/phatbot/.

STEWART, J. 2006. Spamthru trojan analysis. Tech. rep., SecureWorks. http://www.secureworks.com/cyber-threat-intelligence/threats/spamthru/.

STEWART, J. 2009. Sinit p2p trojan analysis. Tech. rep., SecureWorks. http://www.secureworks.com/research/threats/sinit/.

STEWART, J. 2010. Zeus banking trojan report. Tech. rep., SecureWorks. http://www.secureworks.com/cyber-threat-intelligence/threats/zeus/.

STONE-GROSS, B., COVA, M., CAVALLARO, L., GILBERT, B., SZYDLOWSKI, M., KEMMERER, R., KRUEGEL, C., AND VIGNA, G. 2009. Your botnet is my botnet: Analysis of a botnet takeover. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09).* ACM Press, New York, 635–647.

STONE-GROSS, B., COVA, M., GILBERT, B., KEMMERER, R., KRUEGEL, C., AND VIGNA, G. 2011. Analysis of a botnet takeover. *IEEE Secur. Privacy 9*, 1, 64–72.

STOVER, S., DITTRICH, D., HERNANDEZ, J., AND DIETRICH, S. 2007. Analysis of the storm and nugache trojans: P2P is here. *USENIX 32*, 6, 46–63.

STRAYER, W., LAPSELY, D.,WALSH, R., AND LIVADAS, C. 2008. Botnet detection based on network behavior. In *Botnet Detection. Advances in Information Security Series*, vol. 36, Springer, 1–24.

SYMANTEC. 2008. Symantec global internet security threat report, trends for july- december 07. Tech. rep. http://eval.symantec.com/mktginfo/enterprise/white papers/b-whitepaper_internet_security_threat_report_ xiii_04-2008.en-us.pdf.

SYMANTEC. 2010. Symantec global internet security threat report trends of 2009. Tech. rep. DIY kit of Turkojan, Symantec. TURKOJAN. http://www.turkojan.com/eng/.

VAN DER MERWE, A., LOOCK, M., AND DABROWSKI, M. 2005. Characteristics and responsibilities involved in a phishing attack. In *Proceedings of the 4th International Symposium on Information and Communication Technologies (WISICT'05).* 249–254.

VILLAMARIN-SALOMON, R. AND BRUSTOLONI, J. C. 2008. Identifying botnets using anomaly detection techniques applied to DNS traffic. In *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC'08).* 476–481.

WANG, P., SPARKS, S., AND ZOU, C. 2010a. An advanced hybrid peer-to-peer botnet. *IEEE Trans. Dependable Secure Comput. 7*, 2, 113–127.

WANG, P., WU, L., ASLAM, B., AND ZOU, C. C. 2009a. A systematic study on peer-to-peer botnets. http://www.eecs.ucf.edu/~czou/research/P2P-Botnet-ICCCN09.pdf.

WANG, P., WU, L., CUNNINGHAM, R., AND ZOU, C. C. 2010b. Honeypot detection in advanced botnet attacks. *Int. J. Inf. Comput. Secur. 4*, 30–51.

WANG, W., FANG, B., ZHANG, Z., AND LI, C. 2009b. A novel approach to detect IRC-based botnets. In *Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing.* Vol. 1. 408–411.

WEBER, T. 2007. Criminals may overwhelm the web. Tech. rep., BBC News. http://news.bbc.co.uk/2/hi/business/6298641.stm.

WILBUR, K. C. AND ZHU, Y. 2009. Click fraud. *Market. Sci. 28*, 293–308.

WILSON, C. 2007. Botnets, cybercrime, and cyberterrorism: Vulnerabilities and policy issues for congress. Tech. rep., CRS Report for Congress. http://www.fas.org/sgp/crs/terror/RL32114.pdf.

WURZINGER, P., BILGE, L., HOLZ, T., GOEBEL, J., KRUEGEL, C., AND KIRDA, E. 2009. Automatically generating models for botnet detection. In *Proceedings of the 14th European Conference on Research in Computer Security (ESORICS'09).* Springer, 232–249.

XIE, Y., YU, F., ACHAN, K., PANIGRAHY, R., HULTEN, G., AND OSIPKOV, I. 2008. Spamming botnets: Signatures and characteristics. In *Proceedings of the ACM SIGCOMM Conference on Data Communication (SIGCOMM'08).* 171–182.

YADAV, S., REDDY, A. K. K., REDDY, A. N., AND RANJAN, S. 2010. Detecting algorithmically generated malicious domain names. In *Proceedings of the 10th Annual Conference on Internet Measurement (IMC'10).* ACM Press, New York, 48–61.

YADAV, S. AND REDDY, A. N. 2011. Winning with DNS failures: Strategies for faster botnet detection. In *Proceedings of the 7th International ICST Conference on Security and Privacy in Communication Networks (SecureComm'11).*

YAN, W., ZHANG, Z., AND ANSARI, N. 2008. Revealing packed malware. *IEEE Secur. Privacy 6*, 5, 65–69.

YU, F., XIE, Y., AND KE, Q. 2010a. Sbotminer: Large scale search bot detection. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM'10).* 421–430.

YU, X., DONG, X., YU, G., QIN, Y., AND YUE, D. 2010b. Data-adaptive clustering analysis for online botnet detection. In *Proceedings of the 3rd International Joint Conference on Computational Science and Optimization (CSO'10).* Vol. 1. 456–460.

ZEIDANLOO, H., SHOOSHTARI, M., AMOLI, P., SAFARI, M., AND ZAMANI, M. 2010. A taxonomy of botnet detection techniques. In *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT'10).* Vol. 2, 158–162.

ZENG, Y., HU, X., AND SHIN, K. G. 2010. Detection of botnets using combined host- and network-level information. In *Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'10).* 291–300.

ZETTER, K. 2009. Trick or tweet? Malware abundant in twitter urls. Tech. rep., Kaspersky. http://www.wired.com/threatlevel/2009/10/twitter malware/.

ZHANG, J., PERDISCI, R., LEE, W., SARFRAZ, U., AND LUO, X. 2011. Detecting stealthy p2p botnets using statistical traffic fingerprints. In *Proceedings of the 41st IEEE/IFIP International Conference on Dependable Systems Networks (DSN'11).* 121–132.

ZHAO, Y., XIE, Y., YU, F., KE, Q., YU, Y., CHEN, Y., AND GILLUM, E. 2009. BotGraph: Large scale spamming botnet detection. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI'09).* USENIX Association, Berkeley, CA, 321–334.

ZHU, Z., LU, G., CHEN, Y., FU, Z., ROBERTS, P., AND HAN, K. 2008. Botnet research survey. In *Proceedings of the 32nd Annual IEEE International Computer Software and Applications (COMPSAC'08).* 967–972.

ZHUGE, J., HOLZ, T., SONG, C., GUO, J., HAN, X., AND ZOU, W. 2008. Studying malicious websites and the underground economy on the chinese web. In *Proceedings of the Workshop on the Economics of Information Security (WEIS'08).*

Received July 2011; revised July 2012; accepted July 2012