International Journal of Marketing and Technology

CONTENTS

Editorial Advisory Board

Associate Editors

Reviewers

Title

BOTNETS: LIFECYCLE, ATTACKS, DETECTION AND PREVENTION

Author(s)

Mriga Gupta

M.Tech Student of Computer Science & Engineering

Shaheed Bhagat Singh College of Engineering and Technology

Ferozepur, Punjab, India.

Abstract:

Botnets are arguably the biggest threat that the Internet community has faced. The prevalence of botnets, which is defined as a group of infected machines, have become the predominant factor among all the internet malicious attacks such as DDoS, Spam, and Click fraud. In this paper, a survey of botnets is provided. We first discuss fundamental concepts of botnets, including lifecycle, and two major kinds of topologies such as IRC based protocols and P2P based bots. Several related attacks, detection, tracing, and countermeasures, are then introduced, followed by possible future challenges. In order to better understand the challenges that the security community faces in order to dismantle botnets, we first need to understand how botnets function, and the many tools and techniques employed by them. The major objective of this paper is to exploit open issues in botnet detection and preventive measures through exhaustive analysis of botnets features and existing researches.

Keywords: Bot, Botmaster, Honeypot, IRC-based botnets, P2P botnets, Honeynets.

Introduction:

Botnets are emerging threats with billions of hosts worldwide infected. A botnet is an army of compromised machines, also known as "zombies" [10]. Under a command and control(C&C) infrastructure, botnets are able to form a self-propagating, self-organizing, and autonomous framework. Generally, to compromise a series of systems, the botmaster (also called as perpetrator) will remotely control bots to install worms, Trojan horses, or backdoors on them [6]. Currently, honeynets and Intrusion Detection System (IDS) are two major techniques to prevent their attacks. Honeynets are capable of providing botnet attacking information [2]. The IDS uses the signatures or behavior of existing botnets for reference to detect potential attacks.

Botnet Life Cycle:

The general life cycle of a botnet, shown in Figure 1, contains four phases: initial infection, secondary injection, maintenance and update, and malicious activities [13].

a) Initial Infection: A computer can be infected in different ways: Inadvertently execute malicious code, exploit system vulnerabilities, and access through engineered backdoors [12]. Users may accidentally download and execute the malicious programs while viewing a Web Site, opening an attachment from an email, or clicking a link in an incoming instant message [12]. Every released patch to update some of the most popular operating systems, such as Windows XP and Windows 7, is followed by a flurry of reverse engineering in the hacker community in order to exploit the problems that the most recent patch has fixed, because millions of users tend not to update their computer promptly and properly [5,12].

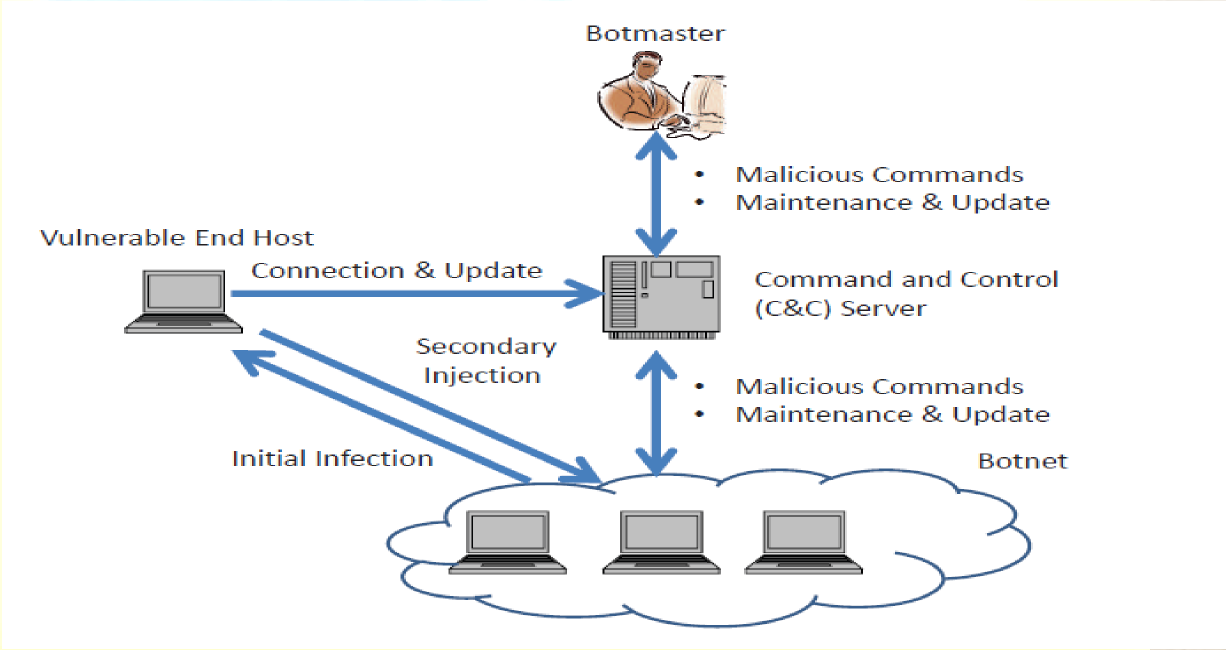

Figure 1: General Botnet Life Cycle [12].

b) Secondary Injection: After the successful initial infection, the next step is to download and run the botnet code in order to become a bot which is under control of a specific botmaster [12]. This procedure can be processed by using Trivial File Transfer Protocol (TFTP), File Transfer Protocol (FTP), HyperText Transfer Protocol (HTTP) or CSend [6].

c) Maintenance and Update: The first two stages only contain communications between bots and targeted computer. After becoming a bot, the infected machine starts to 1) log into the command and control server and 2) create a protected session parsing and executing the topics in the channel [12]. These two steps are processed periodically and require authentication. Before the botmaster authorizes certain malicious activities, such as Distributed Denial of Services (DDoS), it usually sends out an update command to the C&C server which in turn contacts the bots to give the botmaster an updated status feedback of the botnet [14].

d) Malicious Activities: Botnets are mostly used for criminally motivated activities which include Distributed Denial of Services, Click Fraudulence, Spamming, Information Leakage and Identity Fraud [12].

1) DDoS Attacks: Botnets are often used for DDoS attacks, which can disable the network services of victim system by consuming its bandwidth, overload the computational resources of the victim system, or even congest the general internet traffic to make some public massive damages [15]. Most commonly implemented attacks by botnets are TCP SYN and UDP flooding attacks [3]. Such attacks are sometimes accompanied by extortion demands. General countermeasure against DDoS attacks requires: (1) controlling a large number of compromised machines; (2) disabling the remote control mechanism [3]. However, more efficient ways are still needed to avoid this kind of attack.

2) Click Fraudulence: Instead of attacking a web site at the same time, bots are controlled to automatically and periodically access particular links to artificially increase the number of clicks or manipulate the outcomes of online polls [12]. Because each victim's host owns a unique IP address scattered across the globe, every single click will be regarded as a valid action from a legitimate person [5].

3) Spamming: About 70% to 90% of the world's spam is caused by botnets nowadays [16, 17]. An IronPort study in June 2006 estimated that 80 percent of all spam came from botnets, an increase of 30 percent year-over-year for the same period [1].

4) Information Leakage: Some bots may sniff not only the traffic passing by the compromised machines but also the command data within the victims, botmaster can

retrieve sensitive information like usernames and passwords from botnets easily [5]. Since the bots rarely affect the performance of the running infected systems, they are often out of the surveillance area and hard to be caught [6, 18].

5) Identity Fraud: Identity Fraud, also called as Identity Theft, is a fast growing crime on the Internet [12]. Phishing mail is a typical case. It usually includes legitimate-like URLs and asks the receiver to submit personal or confidential information [5]. Botnets also can set up several fake websites pretending to be an official business sites to harvest victims' information. Once a fake site is closed by its owner, another one can pop up, until you shut down the computer [19].

Botnet History and Trends:

Table 1: History of botnets [5]

Timeline	Bot technology
1988	Invention of IRC by Jarkko “WiZ” Oikarinen of the University of Oulu, Finland
1989	Greg Lindahl invents GM the first Bot, where GM plays "Hunt the Wumpus" with IRC users
1999	Pretty Park discovered. First worm to use an IRC server as a means of remote control
1999	SubSeven Trojan/bot. A remote control Trojan added control via IRC
2000	GT Bot, mIRC based. Runs scripts in response to IRC server events Supports raw TCP and UDP Socket connections
2002	SDBot. Written in C++ where its source code is available to hacker community though a small single binary
2002	AgoBot, Gaobot. They introduced modular design. The 1st module break-sin downloads; the 2nd module turns off anti virus and hides from detection before

	downloading the 3rd module. Module 3 has attack engines/payload
2003	SpyBot. Spyware capabilities (key logging, data mining for email addresses lists of URLs, etc.)
2003	RBot. Most Prevalent Bot today. It spreads through weak passwords, easily modifiable, Uses packaging software
2004	PolyBot. A derivative of AgoBot with Polymorphic ability. Changes the look of its code on every infection
2005	MYTOB. My Doom mass emailing worm with Bot IRC C&C

IRC-based protocols:

IRC has provided a common protocol for text-based instant messaging among people that is widely deployed across the Internet for activities among large number of machines, such as remote control and data distribution [20]. IRC has a simple text based command syntax which makes it flexible to be extended for custom functionalities. These features have made IRC the most suitable choice for a botmaster because IRC provides a simple, low-latency, widely available and anonymous command and control channel for botnet communication. Major parts of a typical IRC bot attack are showed in Figure 2 [21].

a) Bot is typically an executable file triggered by a specific command from the IRC server. Once a bot is installed on a victim host, it will make a copy into a configurable directory and let the malicious program to start with the operating system [21].

b) Control channel is a secured IRC channel set up by the attacker to manage all the bots [5].

c) IRC Server may be a compromised machine or even a legitimate provider for public service [5].

d) Attacker is the one who control the IRC bot attack [5].

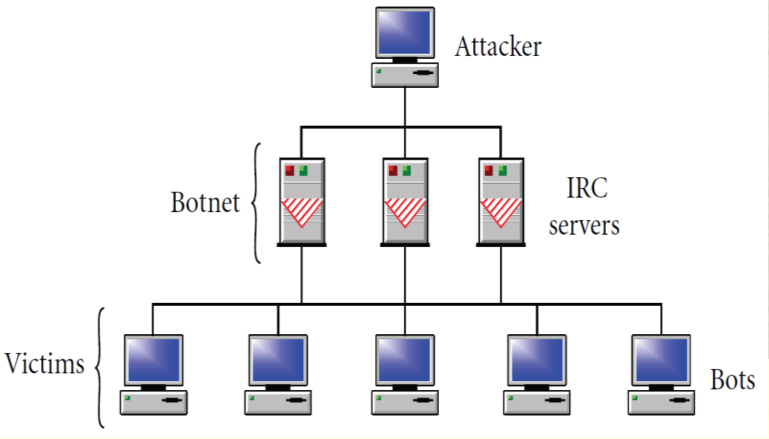

Figure 2: Major parts of a typical IRC Bot attack [5].

P2P based Bots:

In P2P architecture, peer bots act as both clients and servers such that there is no centralized coordination point that can be incapacitated [12]. Because of the lack of the central server, the botmaster cannot directly control all the bots [12]. A worm with a P2P fashion, named Slapper [25], infected Linux system by DoS attack in 2002. The lack of encryption implementation and command authentication has made Slapper vulnerable to be hijacked by others, therefore, hard to be monitored [12]. One year after, another P2P-based bot appeared, called Dubbed Sinit [26]. Later, in 2004, Phatbot [27] were created to send commands to other compromised hosts using a P2P system. Currently, Conficker [28] has its C&C channel encrypted with the most sophisticated algorithms, and the list of possible C&C server Domain names/IP addresses are around 5000 updated on a daily basis.

Types of Bots:

Many types of bots in the network have already been discovered and studied [6, 7, 18]. Some typical types are as follows.

a) Agobot: It is named after its creator Ago, was first released in C++ in 2002 [6]. It is the only bot that can use other control protocols besides IRC. It offers various approaches to hide bots on the compromised hosts [12].

b) SDBot: It was originally written in C and released by a Russian programmer known as sd [6]. It has no more than 2500 lines. Different from Agobot, its code is unclear and only has limited functions. Even so, this group of bots is still widely used in the Internet [18].

c) SpyBot: It first emerged in 2003 [12]. It is written in C with no more than 3,000 lines [5]. There are hundreds of variants of Spybot nowadays [7]. Besides the essential command language implementation, it also involves the scanning capability, host control function, and the modules of DDoS attack and flooding attack, but it does not provide accountability or conceal their malicious purpose in codebase [7].

d) GT Bot: GT (Global Threat) Bot is mIRC-based bot. It enables a mIRC chat-client based on a set of binaries (mainly DLLs) and scripts. It often hides the application window in compromised hosts to make mIRC invisible to user [5]. Based on the limited capabilities in GT Bot, it appears that different versions have been generated for specific malicious intent, instead of general enhancement of the code to provide a broad set of capabilities [41].

Botnet Detection:

Along with the prevalence of botnets related nefarious activities, increasing numbers of botnet detection and tracking techniques have been developed in recent years. These methods can be categorized into two approaches. One is honeynet based method and the other is based on passive traffic monitoring.

a) Honeynet-based Methods: The general structure of honeynet based method consists of honeypot and honeywall [6]. Honeypot denotes end hosts which are well-known by their strong ability to detect security threats, collect malwares, and to understand the behaviors and motivations of botmasters [5]. Honeywall denotes software which is used to monitor, collect, control, and modify the traffic through the honeypot, such as Snort [12]. The Honeynet project used unpatched versions of Windows 2000 or Windows XP systems as honeypot, and snort_inline as honeywall device to track botnets on a daily basis [12]. Honeynet, for monitoring a large-scale diverse network, consists of more than one honeypot on a network. Most of researchers focus on Linux-based honeynet, due to the obvious reason that, compared to any other platform, more freely honeynet tools are available on Linux [36]. As honeypots have become more and more popular in monitoring and defense systems, intruders begin to seek a way to avoid honeypot traps [30]. There are some feasible techniques to detect honeypots. For instance, to detect VMware or other emulated virtual machines [31, 32], or, to detect the responses of program's faulty in honeypot [33]. In [34], Bethencourt et al. have successfully identified honeypots using intelligent probing according to public report statistics. In addition, Krawetz [35] have presented a commercial spamming tool capable of anti-honeypot function, called "Send-Safe's Honeypot Hunter." By checking the reply form remote proxy, spammer is able to detect honeypot open proxies [35]. However, this tool cannot effectively detect others except open proxy honeypot. Recently, Zou and Cunninqham [30] have proposed another methodology for honeypot detection based on independent software and hardware. In their paper, they also have introduced an approach to effectively locate and remove infected honeypots using a P2P structured botnet [30].

b) Passive Traffic Monitoring: This approach is based on setting up vantage points to passively monitor the real Internet traffic and to detect or extract the botnet related packets [12]. Based on different types of Internet traffic data, such as DNS data, BGP route views, Net flow data, and proprietary enterprise data, and on the complexity and response time requirements, many Intrusion Detection System (IDS) designs have been proposed [12]. These techniques can be classified as behavior-based, DNS-based as described and summarized in the following sections [12].

1) Behavior-based Detection: Behavior based detection methods can be further categorized as signature based and anomaly based [12].

a) Signature-based Detection: The knowledge of useful signatures of existing and captured botnets has provided great guidance in botnet detection but they are limited to detect only the known botnets [12]. One of the main problems using signature-based IDS is certainly the maintenance of the signature database. The signatures have to be updated very regularly and the generation of new signatures for the detection of new attacks has to be efficient and, if possible, real-time. This is particularly important for the handling of zero-day attacks [42]. As soon as a new signature is available, the database should be updated, otherwise, the system becomes needlessly vulnerable. For example, Snort [32] is an open source IDS that monitors network traffic to find signs of intrusion by searching matches based on the predefined set of rules and signatures. A major weakness of the signature based detections is that they are limited to detect only the known botnets.

b) Anomaly-based Detection: Different from normal internet traffic, botnets often generates high volume of traffic that may cause high network latency, and traffic on unusual ports [12]. These network traffic anomalies along with other unique botnet behaviors have been utilized for botnet detection [12]. This method combines IRC tokenization and IRC message statistics with TCP-based anomaly detection module [10]. It collects information of large number of TCP packets with respect to IRC hosts [10]. Based on the ratio computed by the total amount of TCP control packets over total number of TCP packets, it is able to detect some anomaly activities [10]. They called this ratio as the TCP work weight and claimed that high value implied a potential attack by a scanner or worm [10]. However, this mechanism may not work if the IRC commands have been encoded, as discussed in [10].

2) DNS-based Detection: For a botmaster to maintain and hide its bots, DNS queries have been implemented in multiple botnet stages, such as the rallying process after infection, malicious attack initiation, and C&C server update. Since bots usually send DNS queries in order to access the C&C servers, if we can intercept their domain names, the botnet traffic is able to be captured by blacklisting the domain names [37, 38]. There are two major factors to distinguish botnet DNS queries from legitimate DNS queries. A first weakness is that queries to C&C servers come only from botnet members; only the bots will send DNS queries to the domain of C&C servers, a legitimate one never do this. Dagon [38] has proposed a mechanism to identify the domain names of the C&C servers with abnormally high or temporally concentrated DDNS query rates. Schonewille and van Helmond [40] found that when C&C servers had been taken down, DDNS would often response name error. Hosts who repeatedly did such queries could be infected and

thus to be suspected [40]. In [39], authors evaluated the above two methods through experiments on the real world. They claimed that, Dagon's approach was not as effective since it misclassified some C&C server domains with short TTL, while Schonewille's method was comparatively effective due to the fact that the suspicious name came from independent individuals [39]. A second weakness is that bots usually generate highly correlated DNS queries. Choi et al. [37] proposed a botnet detection mechanism that monitors group activities which are often consist of DNS queries simultaneously sent by a large number of distributed bots. This method is more robust than the aforementioned two and is botnet-type independent [12]. Furthermore, it can also detect botnets with encrypted channels since it uses information in IP headers [12]. The main drawback of this approach is the high processing time required for detailed monitoring of the huge scale of network traffic [12].

Preventive Measures:

It takes only a couple of hours for conventional worms to circle the globe since its release from a single host [5]. If worms using botnet appear from multiple hosts simultaneously, they are able to infect the majority of vulnerable hosts worldwide in minutes [2]. Botnets are problematic for a number of reasons: 1) We have no idea how many botnets are out there. 2) We have no idea how big the active botnets are. 3) Size is not correlated directly to lethality. 4) Many botnets are programmable. 5) Bots create a lot of 'network noise' as they scan and attack other hosts. In order to minimize the risk caused by botnets, certain preventive measures are to be kept in mind.

a) Countermeasures on Botnet Attacks: There are very few solutions in existence for a host to detect botnet attacks [1]. Although it is hard to find the patterns of malicious hosts, various network administrators can still identify botnet attacks based on passive operating system fingerprinting extracted from the latest firewall equipment [1]. The lifecycle of botnets tells us that bots often utilize free DNS hosting services to redirect a sub-domain to an inaccessible IP address. Thus, removing those services may take down such a botnet [1]. At present, many security companies focus on offerings to stop botnets [1]. Some of them protect consumers, whereas most others are designed for ISPs or enterprises.

b) Countermeasures for Public: Personal or corporation security inevitably depends on the communication partners [2]. Firstly, one should continuously request the service supplier for security packages, such as firewall, anti-virus tool-kit and intrusion detection utility [2]. Once something goes wrong, there should be a corresponding contact number to call [2]. Secondly, one should also pay much attention on network traffic and report it to ISP if there is a DDoS attack [2]. ISP can help blocking those malicious IP addresses [2]. Thirdly, it is better to establish accountability on its system, together with law enforcement authority [2]. Scholars and industries have proposed some strategies for both home users and system administrators, to prevent, detect and respond botnet attacks [18, 21].

1) Home Users. To prevent attacks from a botnet, home users can follow the rules described in Table 2 [5]. These are classified into three categories: (1) Personal Habits, (2) Routine, and (3) Optional Operations [5]. As personal habits, people should pay attention when downloading, especially for those programs coming from unscrupulous sites [5]. Besides, try to avoid installing useless things on personal computer, which will minimize the possibility of bots infection [5]. If necessary, read the License Agreement and the notes carefully before click the button on the web site [5]. As a routine, use antivirus software and anti-Trojan utilities while system is on. Scan and update system regularly, especially for Windows [5]. When leaving the PC, shutdown the system or it may be remotely controlled by hackers [5]. As the optional operations, home users are recommended to backup system regularly, to keep all software up-to-date and to deploy personal firewall by all means [5]. By doing so, home PCs are shielded from unauthorized accesses, and thus bots cannot compromise them. If unusual behavior occurs on a home PC, such as slow network response, unknown ports being used, and something like that, there is possibly a bot attack [18, 21]. Also, home users can use anti-virus software or online services to detect attacks [18, 21]. Once the computer has been compromised, there are strategies to recover it.

2) System Administrator. Similarly, there are corresponding rules for system administrators to prevent, detect, and respond botnet attacks [18, 21]. For a prevention method, system administrators should follow vendor guidelines for updating the system and applications [21]. Also, keep informed of latest vulnerabilities and use access control and log files to achieve accountability [21]. As illustrated in Table 3, the following measures can help the system administrator to minimize the possibilities of botnet attacking. Once an attack is detected, a system administrator should isolate those compromised hosts and notify the home users [18]. Then preserve the data on those infected hosts including the log files [18]. Besides, identify the number of victims via sniffer tools [18]. Finally, report the infection to security consultant [18].

Table 2: Rules of prevention by home users [5]

Type	Strategies
Personal Habits	Attention while downloading Avoid installing useless things Read carefully before you click
Routine	Use anti-virus/ Trojan utilities Update system frequently Shutdown PC when you have
Optional Operations	Back-up all systems regularly Keep all software up-to-date Deploy personal firewall

Table 3: Rules of Detection by System administrators [5]

Rules	Notes
Monitor logs regularly	Analyze the internet traffic for anomalies
Use network packet sniffer	Identify the malicious traffic in internet
Isolate the malicious subnet	Verify IRC activity on host
Scan individual machine	They may contain malware

Conclusion and Future Challenges:

Bots have grown from simple tools to automate tasks on IRC to a major threat to the Internet and the companies and institutions that use the internet for commercial, educational and scientific benefit. This threat also attacks the home users, denying them access to site they want to use, or using their own systems to perform these attacks; steal their data, their identities, use their systems to store stolen or illegal material, to route spam, malware or scams through. Detecting and tracking compromised hosts in a botnet will continue to be a challenging task. Since 1989, botnets have evolved from the benign assistant tool to the predominant threat in modern internet. Although the number of bots to each botnet seems to be decreasing, the monetary damaging power of the botnets is continuously increasing given the development of internet bandwidth. We need an up-to-date knowledge base for all released bots in the world, which seems to be an impossible mission. Since current detecting technology depends on the happened attacking event, no guarantee for us to find every possible compromised hosts. Instead of using a centralized, IRC based C&C channel to perform multiple nefarious attacks, the botnets have been gradually developed into more complicated, stealthy, and modular based package which perform particular malicious activity with diverse C&C protocols and structures. There is no doubt that we will see other uses for bot infected systems in the near future.

References:

- Wikipedia, “Botnet,” http://en.wikipedia.org/wiki/Botnet.
- G. P. Schaffer, “Worms and viruses and botnets, oh my!:rational responses to emerging internet threats,” IEEE Security and Privacy, vol. 4, no. 3, pp. 52–58, 2006.
- F. C. Freiling, T. Holz, and G. Wicherski, “Botnet tracking: exploring a root-cause methodology to prevent distributed denial-of-service attacks,” in Proceedings of the 10th European Symposium on Research in Computer Security (ESORICS ’05), vol. 3679 of Lecture Notes in Computer Science, pp. 319–335, Springer, Milan, Italy, September 2005.
- H. Choi, H. Lee, H. Lee, and H. Kim, “Botnet detection by monitoring group activities in DNS traffic,” in Proceedings of the 7th IEEE International Conference on Computer and Information Technology (CIT ’07), pp. 715–720, Fukushima, Japan, October 2007.
- Jing Liu, Yang Xiao, Kaveh Ghaboosi, Hongmei Deng, and Jingyuan Zhang, “Botnet: Classification, Attacks, Detection, Tracing, and Preventive Measures,” EURASIP Journal on Wireless Communications and Networking, Volume 2009, Article ID 692654.
- P. Bacher, T. Holz, M. Kotter, and G. Wicherski, “Know your enemy: Tracking botnets.” http://www.honeynet.org/papers/bots, 2005.
- P. Barford and V. Yegneswaran, “An inside look at botnets. Malware Detection,” pages 171–191, 2006.
- D. Barr. RFC 1912: Common DNS operational and configuration errors. http://www.ietf.org, Feb. 1996. Obsoletes RFC1537 [6]. Status: INFORMATIONAL.
- P. Beertema. RFC 1537: Common DNS data file configuration errors. http://www.ietf.org, Oct. 1993 obsolete by RFC1912 [5]. Status: INFORMATIONAL.
- J. Binkley and S. Singh, “An algorithm for anomaly-based botnet detection,” in Proceedings of USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI), pages 43–48, 2006.
- M. Masud, T. Al-khateeb, L. Khan, B. Thuraisingham, and K. Hamlen, “Flow-based identification of botnet traffic by mining multiple log files.” in Distributed Framework and Applications, First International Conference on, pages 200–206, 2008.

- Xiaonan Zang, Athichart Tangpong, George Kesidis and David J. Miller, "Botnet Detection Through Fine Flow Classification," CSE Dept Technical Report No. CSE11-001, Jan. 31, 2011.
- E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: Understanding, detecting, and disrupting botnets," in Proceedings of the USENIX SRUTI Workshop, pages 39–44, 2005.
- Intrusion Detection Systems FAQ, WindowSecurity.com, see: http://www.windowsecurity.com/articles/Intrusion_Detection_FAQ.html
- Jelena Mirkovic, Janice Martin and Peter Reiher, "A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms," University of California, Los Angeles Technical report #020018.
- K. Pappas, "Back to basics to fight botnets," Communications News, vol. 45, no. 5, p. 12, 2008.
- P. Sroufe, S. Phithakkitnukoon, R. Dantu, and J. Cangussu, "Email shape analysis for spam botnet detection," in Proceedings of the 6th IEEE Consumer Communications and Networking Conference (CCNC '09), pp. 1–2, Las Vegas, Nev, USA, January 2009.
- J. Govil, "Examining the criminology of bot zoo," in Proceedings of the 6th International Conference on Information, Communications and Signal Processing (ICICS '07), pp. 1–6, Singapore, December 2007.
- "Top 10 Botnet Threat Report – 2010" Damballa Inc. http://www.damballa.com/downloads/r_pubs/Damballa_2010_Top_10_Botnets_Report.pdf
- J. Oikarinen and D. Reed, RFC 1459: Internet Relay Chat Protocol. http://www.ietf.org, 1993.
- R. Puri, "Bots and botnets: an overview," Tech. Rep., SANS Institute, 2003.
- T. Holz, M. Steiner, F. Dahl, E. W. Biersack, and F. Freiling, "Measurement 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, pp. 1–9, San Francisco, Calif, USA, April 2008.
- P. Wang, S. Sparks, and C. C. Zou, "An advanced hybrid peer-to-peer botnet," in Proceedings of the 1st Workshop on Hot Topics in Understanding Botnets, p. 2, Cambridge,Mass, USA, July 2008.

- R. Lemos, “Bot software looks to improve peerage,” http://www.securityfocus.com/news/11390.
- I. Arce and E. levy, “An analysis of the slapper worm,” IEEE Security & Privacy Magazine, vol. 1, no. 1, pp. 82–87, 2003.
- J. Stewart, “Sinit P2P Trojan analysis,” http://www.secureworks.com/research/threats/sinit/.
- J. Stewart, “Phatbot Trojan analysis,” http://www.secureworks.com/research/threats/phatbot/?threat=phatbot.
- M. Bowden. The enemy within. http://www.theatlantic.com/magazine/archive/2010/06/the-enemy-within/8098/, June 2010.
- J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, “Globally distributed content delivery,” IEEE Internet Computing, pages 50–58, 2002.
- C. C. Zou and R. Cunningham, “Honeypot-aware advanced botnet construction and maintenance,” in Proceedings of the International Conference on Dependable Systems and Networks (DSN ’06), pp. 199–208, Philadelphia, Pa, USA, June 2006.
- J. Corey, “Advanced honey pot identification and exploitation,” 2004, http://www.ouah.org/p63-0x09.txt.
- K. Seifried, “Honeypotting with VMware basics,” 2002, http://www.seifried.org/security/index.html.
- Honeyd security advisory 2004–001, “Remote detection via simple probe packet,” 2004, http://www.honeyd.org/adv.2004-01.asc.
- J. Bethencourt, J. Franklin, and M. Vernon, “Mapping internet sensors with probe response attacks,” in Proceedings of the 14th Conference on USENIX Security Symposium, pp. 193–208, Baltimore, Md, USA, August 2005.
- N. Krawetz, “Anti-Honeypot technology,” IEEE Security and Privacy, vol. 2, no. 1, pp. 76–79, 2004.
- B. McCarty, “Botnets: big and bigger,” IEEE Security and Privacy, vol. 1, no. 4, pp. 87–90, 2003.
- H. Choi, H. Lee, H. Lee, and H. Kim, “Botnet detection by monitoring group activities in DNS traffic,” in Proceedings of the 7th IEEE International Conference on Computer and Information Technology (CIT ’07), pp. 715–720, Fukushima, Japan, October 2007.

- D. Dagon, “Botnet detection and response, the network is the infection,” 2005, http://www.caida.org/workshops/dnsoarc/200507/slides/oarc0507-Dagon.pdf.
- R. Villamarin-Salomon and J. C. Brustoloni, “Identifying botnets using anomaly detection techniques applied to DNS traffic,” in Proceedings of the 5th IEEE Consumer Communications and Networking Conference, pp. 476–481, Las Vegas, Nev, USA, January 2008
- A. Schonewille and D. J. van Helmond, “The domain name service as an IDS,” M.S. thesis, University of Amsterdam, Amsterdam, The Netherlands, February 2006.
- Paul Barford and Vinod Yegneswaran, “An Inside Look at Botnets,” Computer Sciences Department, University of Wisconsin, Madison
- What’s a “zero-day” attack? http://ask-leo.com/whats_a_zeroday_attack.html