ISSN No. 0976-5697

Volume 4, No. 6, May 2013 (Special Issue)

International Journal of Advanced Research in Computer Science

REVIEW ARTICAL

Available Online at www.ijarcs.info

Botnet Analysis and Detection

Prof Amit sahu
G.H.Raisoni College of Engineering and technology
Dept of computer science and Engineering
Amravati, India
amitsahu@gmail.com

B. G. Dalvi
Dr. P.D.Polytechnic
Dept of computer science and Engineering
Amravati, India
dalvi.bharatirediffmail.com

Abstract— Among the various forms of malware, botnets are emerging as the most serious threat against cyber-security as they provide a distributed platform for several illegal activities such as launching distributed denial of service attacks against critical targets, malware dissemination, phishing, and click fraud. The defining characteristic of botnets is the use of command and control channels through which they can be updated and directed. Recently, botnet detection has been an interesting research topic related to cyber-threat and cyber-crime prevention. This paper is a survey of botnet and botnet detection. The survey clarifies botnet phenomenon and discusses botnet detection techniques. This survey classifies botnet detection techniques into four classes: signature-based, anomaly-based, DNS-based, and mining-base. It summarizes botnet detection techniques in each class and provides a brief comparison of botnet detection techniques.

Keywords: Botnet; Botnet Detection; Cyber-security

I. INTRODUCTION

Now a day Botnets are emerging as serious security threat to the cyber world. Botnets are compromised computers which act as slaves to the master computer, which carry out the malicious activities. Major transmission between master and slave is based on the command and control protocols. There are several kinds of botnets like, IRC botnets, web based botnets, peer to peer botnets, like many types of botnets are floating around to carry out malicious activities. They're targets are to carry out the several tasks like spamming, distributed denial of service attacks, password sniffing, privilege escalation, financial gain, key logging, in present trend they are even using to generate fake traffic to the websites which is also a method of financial gain with popular websites like adsence. Based on the taxonomy of botnets the previous searches in this field say there are mainly three types of topologies which are peer-to-peer, centralized, and random (Zhenqi Wang, 2010).

The big is the botnet causes big damage to the network. The major operations carried out using botnet are DDOS attack so the botnet with thousand of infected computer causes more damage than the small botnets. Traditional botnets works with central command and control system which gives advantage to find the command centre and can takedown entire botnet. In order to overcome disadvantages attackers comes with peer-to-peer method but the method of approach and requirements are quite different. Recent major bot is coreflood virus which is a major security threat in the windows it will open a backdoor Trojan and records keystrokes of the victim. FBI cached this botnet in 14th April 2011. The first major botnet is storm botnet which is detected in September 2007 and with this botnet over 250000 to 1 million computers are infected, although this bot is not very powerful but it caused some serious issues. The bots are different from platform to platform windows bot neither work with Mac pc nor work with Linux pc because of its kernel. Some serious bots make changes to the windows kernel so its existing in victim computer could not reveal to antivirus also they disable the runtime protection as well as scan time protection to be undetected.

In order to conduct the further research this chapter is categorized mainly based on below topics

a. Botnet taxonomy
b. Types of Bots
c. Botnet characteristics and behavior
d. Antivirus mechanism against botnet
e. Analysis and review of different botnets
f. Command and control
g. Botnet detection methods

A. Botnet Taxonomy:

Botnets are becoming major security threats to the cyber world. In order to understand the main aspects of this thesis there are very few words to know those are bot, botnet, IRC, command and control. Bot is a computer which is already infected. Botnet is collection of infected computers or network of infected victims, command and control channel is a communication channel used for transmitting the information between bots and botnet. IRC is internet relay chat which is called a chat program to pass command to bots from bot master. (Botnet: Survey and Case Study, 2009).

B. Botnet evolution:

The threat of botnets is started from 1993 and became very serious and growing very fast over time. The following are the some major botnet findings in these years 1993-2011. The latest major botnet finding is 14/04/2011 which is named as The core flooded virus which causes hundreds of computers are already infected and also caused fraudulent money transfers of thousands of dollars. There are various command and control centers for this botnet (FBI-Botnet Operation Disabled, 2011).

Table: 1

[1] Bot name	[2] Year founded
[3] Egg drop	[4] Dec 1993
[5] GT	[6] April 1998

CONFERENCE PAPER
"A National Level Conference on Recent Trends in Information Technology and Technical Symposium" On 09th March 2013
Organized by
Dept. of IT, Jawaharlal Darda Inst. Of Eng. & Tech., Yavatmal (MS), India

7	Pretty Park	8	June 1999
9	Ago	10	April 2002
11	Slapper	12	September 2002
13	SD	14	October 2002
15	Spy	16	April 2003
17	Sinit	18	September 2003
19	Phatbot	20	March 2004
21	Gaobot	22	March 2004
23	Nugache	24	April 2006
25	Peacomm	26	Jan 2007
27	Kraken	28	April 2008
29	Srizbi	30	July 2008
31	Cutwail	32	Nov 2009
33	Zeus	34	December 2010

The above table represents the most dangerous botnet findings over time. (Botnet: Survey and Case Study, 2009). The cutwail bot sent over 1.7 million spam messages it is based on java script execution which resides inside of pdf file (BitDefender weekly review – The Cutwail botnet. A little insight , 2009). kraken botnet resides in victim system and it will sends out the spam mails . Many operating systems are affected because of this bot and it became very hard for virus companies to find it (dell secure works, 2008). Peacomm will attack with fake names like video.exe and movie.exe like names through email once it will install in system it will open back door Trojan to the server through udp port 4000 and it will use peer to peer connection (Trojan.Peacomm: Building a Peer-to-Peer Botnet, 2009).Nugache bot is different from other bots where it do not connect back to master for commands as it will create p2p network for the commands (W32/Nugache@MM IRC bot, 2006).Gaobot is typical bot as it do not visible in the process list of the computer and upon execution it performs malicious activities like privilege escalation, DDOS attacks, sniffer, CD keys stealing like activities (W32/Gaobot.worm.ali, 2004).sinit bot is called servant bots, these bots do not need boot strappers it communicates via peer list which comes with botnet (paper notes on hybrid p2p bot, 2007).

There are many botnet which cause much extensive damage but among those the following are discovered as most powerful botnets over time.

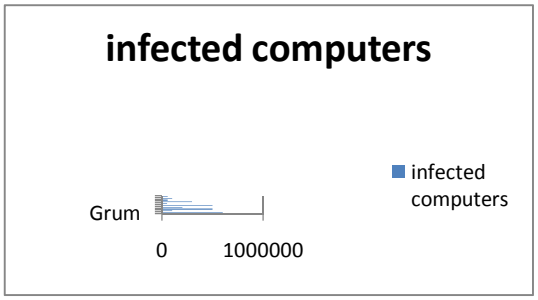

Figure: 1

From the above figure and the research conducted by Daren Lewis symantic employee 80 % of the all spam mails sent up to now are sent by these bots only. Every day these bots sent more than 185 million spam messages. Only for these bot nets there are more than 5 million computers are being infected (The top 10 spam botnets: New and improved, 2010).

C. Types of bots:

Significantly bots are based on many things. Every bot will have each purpose based on the significance of the bot it will design in different interface. Based on the botnet taxonomy bots are mainly divided by following (Nazario, April 27, 2008) Bots by network structure:

Based on the network structure bots are divided into two categories centralized bots and de centralized bots. In which centralized bots are IRC bots and HTTP bots, un centralized bots are p2p bots. In which 90% of the bots are IRC kind, 4% are HTTP bots, 5% bots are p2p bots and remaining bots are 1% (Nazario, April 27, 2008)

a. IRC bots:

IRC is a client master communication system at first developed by Jarkko Oikarinen in 1988. Basically this IRC system is used in chatting system where administrator creates the channel in server. And thus all the clients will join to that particular channel for chatting purpose. IRC botnets are also developed using same principles. IRC bots are very stable but once the bot master is found then it is very easy to take down the entire botnet. IRC bot infected computer bot will join the channel wit randomly generated nickname and it will wait for the commands from the master. And for avoiding the loss by detecting the master admin will keep the multiple chat rooms using dynamic dns system (Zhenqi Wang, 2010).

b. HTTP bots:

Large botnet are controlled by issuing the commands through command and control mechanism of the bots. These have to be issued with the bot master. So in order to keep the bots updated bot master has to give commands. To overcome this problem Http botnets are evolved these bots will contact with the web server as soon as they planted in the victim computer and they randomly connects to the web server to perform the attacks. The web based http bots blend into the http traffic of the victim so it is hard to find these bots (Binbin Wang).

II. HOW BOTNET WORK

Most botnets are designed as distributed-design systems, with the main botnet operator (*botmaster*) issuing instructions directly to a small number of systems. These machines propagate the instructions to other compromised machines, usually via Internet Relay Chat (IRC) [15]. The constituents of a typical botnet include a server program, client program for operation, and the program that embeds itself on the victim's machine (*bot*). All three of these usually communicate with each other over a network and may use encryption for stealth and for protection against detection or intrusion into the botnet control network [16].

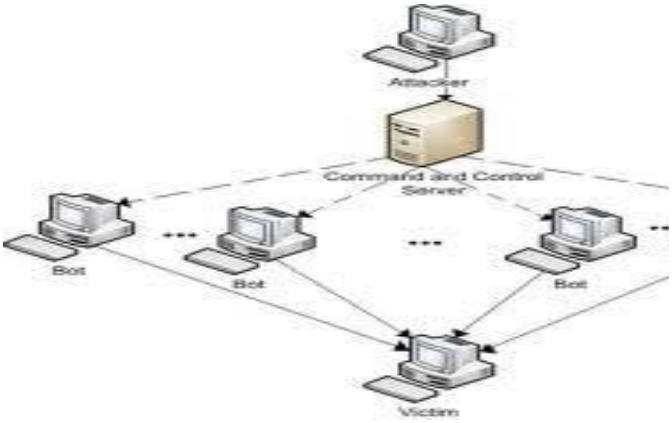

Figure: 2

Fig 2.1 Example of a DDoS Attack [Source: Riverhead Networks] [9] Botnets are effective in performing tasks that would be impossible given only a single computer, single IP address, or a single Internet connection. Originally, botnets were used for distributed denial of service attacks. (See Figure 2.1) Most modern web servers have developed strategies to combat such DDoS attacks, making this use of a botnet less effective [15]. When infecting a computer, the bots connect to IRC servers on a predefined channel as visitors and waited for messages from the botmaster. The botmaster could come online at any time, view the list of bots, send commands to all infected computers at once, or send a private message to one infected machine.

Figure: 3

This is an example of a centralized botnet Fig 2.2 C&C issues commands to Bots.

III. BOTNET ATTACK AND ANALYSIS

Botnets are nothing new to the Internet. Most Internet users have become all too familiar with the near- constant barrage of attacks from all across the world in an attempt to leverage our systems. Botnet herders are in a constant search for new hosts, using any mean necessary to add one more vulnerable server to the swarm, with motives ranging from childish revenge to high-level extortion schemes. Through ouranalysis, I will show how we were able to log an attack as it occurred and also mimic the probable execution of asuccessful attack in order to learn more about a particular botnet. What separates the following attack fromvarious others is the successfully leverage of Google as an enumeration tool to search for vulnerable hosts.

A. Active Exploitation:

In order for a host to become part of a botnet, it first needs to be compromised.The attack we witnessed used a known script-injection vulnerability in the Horde Web Mail Help Module that was released on April 5th, 2006. The HTTP GET request sent to exploit this host resembled:

GET/horde/services/help/?show=about&module=;%22.passthru(%22%20
cd%20%22.chr(47).%22tmp;curl%20O%20x.txt%20http:%22.chr(47).%22
%22.chr(47).%22www.kildarefamily.com%22.chr(47).%22h;wget%20http
:%22.chr(47).%22%22.chr(47).%22www.kildarefamily.com%22.chr(47).
%22h;fetch%20http:%22.chr(47).%22%22.chr(47).%22www.kildarefamil
y.com%22.chr(47).%22h;perl%20h;rm%20-rf%20h%22);'. HTTP/1.1..Acc
ept:*/*..Accept-Language:en-us..Accept-Encoding: gzip, deflate User-Agent: mozilla/4.0(compatible; MSIE 6.0; Windows 98)..Host: 10.99.99.1 ..Connection: Close.

By cleaning up this request, we can see that the injected character translations perform the following commands:

a. 1.cd /tmp;
b. 2.curl-Ox.txt http://www.kildarefamily.com/h;
c. 3.wget http://www.kildarefamily.com/h;
d. 4.fetch http://www.kildarefamily.com/h;
e. 5. perl h;
f. 6. rm -rf h;

There are three attempts to download a Perl script using command-line utilities before running the script and removing it from disk. The script forks itself into the background where it will run in memory until the machine is rebooted. The script was sophisticated enough to setup signal handlers to disregard all signals that could be sent to kill the process.

B. Command and Control:

Since the downloaded Perl script was in plain text with no obfuscation, it was quite easy to determine the purpose and function of the code. Immediately after execution, the script would connect out to an IRC server and then join a specified channel where it would idle while waiting for commands. Each of these connection properties (IRC server address, port, and channel) were hard coded into the script. Delving deeper into the code, the following commands were listed as available to the botnet herders:

portscan <host> - Scan for commonly open ports on a host google <time> <boturl> - Search google for more vulnerable hosts and propagate! tcpflood <host> <time> - Flood a host with TCP connection attempts httpflood <host> <time> - Flood a host with HTTP connections udpflood <host> <time> - Flood a host with UDP Datagrams Direct Command-line Access By far, the most interesting command in this group is the "google" command. When executed, it will search Google using "Google Dorks", a query that reveals mis-configured servers, to find vulnerable hosts indexed through the search engine. The script will directly query Google's search page and then parse the returned data for valid URL's. Then, a character-obfuscated script injection attack string is appended to these URLs and finally queried from the bot with the hope of infecting another host using the same method in which it was infected. Strange as it seems, the vulnerabilities that the script searches for are phpBB related, and not the same Horde Web Mail scripts that attacked this host, yet the attack methodologies are very similar. 1 of 2 8/19/2009 2:25 PM It is not often that a botnet's command-and-control script is readily available to the observant security analyst. I was amazed that a botnet's IRC address, port, and channel

would be so easily accessible to anybody keeping a watchful eye on their network. After obtaining the command-and-control information from the script, sure enough, I was able to connect to this IRC server, join the common Bot channel, and even speak with one of the botnet herders that was idling at the time. Our conversation was very brief since, I imagine, he was not very fond of visitors, especially one asking so many questions. After being kicked from the server twice, he stated that he was happy my web server was not compromised and warned me that they were developing new attacks all the time; shortly thereafter, I was permanently banned from the server. I noted a maximum of fifty seven infected hosts that were continuously joining and parting from the channel with a constant increase in bots over the time I had access.

C. Solution:

There are multiple proactive ways to avert this type of attack. It is always recommended to keep an inventory of web applications running on your web servers, complete with version and patching histories in order to quickly assess if an application is vulnerable to attack. The attack against Horde Web Mail could have easily been averted by applying a patch that was released shortly after the exploit was made public. Also, by only providing the minimal amount of tools to host the web service, the chances of an attacker abusing the capabilities of the server are greatly reduced, as seen with the unauthorized use of curl, fetch, and wget to retrieve the botnet code. Finally, the Bleeding Edge Threats Team have created the "WEB Horde Web Mail Help Access" Snort rule in order to detect and alert on attempts to exploit this service.It is apparent that this Perl-bot script is a variant of older, outdated vulnerability scanners, which leads us to believe that these botnet herders can still teach an old script new tricks.

IV. DETECTING AND TRACKING

There are mainly two approaches of botnet detection andtracking methods. One is honeynet based method and theother is based on passive traffic monitoring.

A. Honeynet:

There are many papers [23, 24] discussed how to trackbotnet using Honeynet, and how to use tools to collect malware[5]. In [22], Jose Nazario from Arbor Networks discussesseveral challenges in developing a botnet trackingcollect malware, but no tool for tracking the botnet. Secondly,the tracking tool needs to understand the botnet's ”jargon” in order to be accepted by the botmaster. Moreover,the increasing use of anti-analysis techniques used bythe blackhat circle makes the development of the tool evenmore challenging.

B. Traffic Monitoring:

In [20] it described a network-wide system to identify botmasters based on transport layer flow information. It gathers traffic flow information from many vantage points within the network. The core idea is based on the attack and control chain of the botnet. The major steps are listed as follows:

a. Identify bots based on their attack activities, such as scanning, emailing of spam and viruses, or DDoS traffic generation. The activities are reported by other security system.
b. Analyze the flows of these bots to find candidate controller connections (CCC).
c. Analyze the CCC to locate the botmaster.

This paper also gives us some interesting results. For example, based on the long-time observation, it estimates the bot stays 2-3 days on the same controller in average. In [14] it presented a passive monitoring system (Rishi) to track botnets based on the bots' IRC nicknames. The core idea is that the format of nicknames used by the bots is different from that of a normal user, e.g. USA|016887436 is a typical nickname used by the bots. The author uses regular expression for the detection. The system is deployed on a border router of a campus network running two weeks, and here are their findings:

a) Results are compared with their NIDS system (Blasto-Mat). 82 bots were detected while only 34 were detected by Blast-o-Mat. Blast-o-Mat detected 20 hosts which were not picked up by Rishi.
b) None of the botnets uses port traditional IRC port 6667 for C&C.

However, this approach is quite limited. For example, IRC Nickname can be changed to resemble normal user. And it can not detect HTTP botnet, or the botnet of which the communication is encrypted, e.g. Rustock mentioned in the following are two more advanced detection tools. A BotHunter system is presented which consists of a correlation engine that is driven by three malware-focused network packet sensors, each charged with detecting specific stages of the malware infection process([17]). It finds the suspicious flows which match BotHunter's infection dialog model. Based on the observation that bots within the same botnet will likely have spatial-temporal correlation and similarity, it proposes using network-based anomaly detection to identify botnet C&C channels([18]).

The most recent work appears in [16]. In this paper presents classifying networking traffic to detect botnet, which is independent of the botnet protocol and structure. 4.1 Defenses Against Botnet Unfortunately, only a few papers proposed defense technologies against botnet. The most effective way is to shutdown the botmaster once we identify it. However, this task is far from trivial. The following discusses the defense and some practical issues with this approach. ASpam In [7] it proposed a distributed, content independent spam classification system to defend from botnet generated spams. A little bit unexpected, the system does not utilize previous botnet detection results to ban emails generated by bots. The basic idea of the system is that ”A host that has recently sent large amounts of e-mails may be a spam-bot. Consequently, any e-mail coming from such hosts is potentially spam, and if the source has a dynamically allocated IP address (or simply a dynamic IP address) and the sender is not in the recipient's address book or list of past recipients or senders, then it is almost certain that the e-mail is spam.”

The system consists of following parts:

a. Identifying the source of emails
b. Keeping track of how many emails were recently sent by a source
c. Disseminating this information for the purposes of classifying future emails.

CONFERENCE PAPER
“A National Level Conference on Recent Trends in Information Technology and Technical Symposium” On 09th March 2013
Organized by
Dept. of IT, Jawaharlal Darda Inst. Of Eng. & Tech., Yavatmal (MS), India

The effectiveness of this system is unknown since it is still in the process of Implement

C. Enterprise Solutions:

Trend Micro provided Botnet Identification Service ([3]). The company provide the customers the real-time botnet C&C botmaster address list via BGP peering between Trend Micro BIS router and the customers' BGP border router. This service charges 9 cents per user for 500,000 users. However, Fast-Flux networks can make Trend Micro's solution much less effective.

V. DISSCUSSION

The previous sections on understanding Botnets Understanding Networks and Understanding Techniques each highlighted the unique challenges faced by today's botnet technology and defenses.

a. Botnets are moving targets. All aspects of the botnet's life-cycle, from propagation, to command and control, and attacks are all evolving constantly. Trying to nail down a specific set of tradeoffs (e.g., survivability verses message latency) or predicting future trends is a losing battle.
b. No technique is perfect. Each detection algorithm or technique comes with its own unique set of tradeoffs with respect to false positives and false negatives and each technique makes a set of assumption about the available insight into the threat and about the aspect of botnet behavior it is discovering.
c. All networks are not the same. Different types of networks (e.g., enterprises, ISPs) approach the botnet problem with differing goals (i.e., notification verse remediation), with different visibility into the botnet behaviors, and different sources of data with which to uncover those behaviors (e.g., network data, host data).

A successful solution for botnet detection and mitigation will need to cope with each of these realities and their complex interactions with each other.

VI. CONCLUSION

While botnets are widespread, the botnet research is still in its infancy. This paper surveys state-of-art botnet research that can be categorized into three areas, i.e. understanding botnet, detecting & tracking botnets, and countering against botnets. In understanding botnet research, it is proposed to learn botnet behaviors and characteristics through source code analysis, binary analysis or wide area measurement. Some formal models are also proposed to predict botnet advancement. In detecting & tracking botnet researches, honeynet and traffic monitoring approaches are proposed to detect botnets based on some of their unique behaviors. Finally, the research on defending against botnet proposes to simply shut down botmaster after they are identified. Those current botnet study is still in a preliminary stage. Previous analysis shows that majority of botnet traditionally used IRC for their command and control. But we believe the botnets will advance to new communication architectures, for example, P2P-based botnet. And currently the defense against botnet is not very efficient, so much more work needs to be done in this field. Finally future botnet prediction may give us an advanced view of the botnet development. Good model can help people know the properties of botnet and thus control it. The following are some topics for possible future work.

VII. REFERENCES

[1]. M. Akiyama, T. Kawamoto, M. Shimamura, T. Yokoyama, Y. Kadobayashi, and S. Yamaguchi. A proposal of metrics for botnet detection based on its cooperative behavior. In Proceedings of the 2007 International Symposium on Applications and the Internet Workshops(SAINTW'07),Washington, DC, May 2007.

[2]. D. S. Anderson, C. Fleizach, S. Savage, and G. M. Voelker. Spamscatter: Characterizing internet scam hosting infrastructure. In Proceedings of the 16th USENIX Security Symposium (Security'07), Boston, MA, August 2007.

[3]. Ferris Research (2009), Industry statistics. Retrieved October 31, 2009 from <http://www.ferris.com/research-library/industry-statistics/>

[4]. Webroot Software, Inc. (2006). From viruses to spyware: In the malwaretrenches with small and medium-sized businesses. Retrieved October31,2009 from http://www.webroot.com/shared/pdf/wp_SMBtrenches.pdf >

[5]. Krebs, B. (February 19, 2006). Invasion of the computer snatchers.

[6]. *Washington Post*. Retrieved October 31, 2009 from http://www.washingtonpost.com/wpdyn/content/article/2006/02/14/AR2006021401342.html

[7]. Nazario, Dr. J. (May 2004). The zombie roundup: Understanding, detecting, and disrupting botnets. Retrieved October 31, 2009from http://www.eecs.umich.edu/~emcooke/pubs/botnets-sruti05.pdf>

[8]. Trend Micro 2007 threat report and forecast. Retrieved October 31, 2009 from <http://trendmicro.mediaroom.com/index.php?s=65&item=163>

[9]. The Honeynet Project. (November 7, 2007). Know your enemy: Behind the Scenes of Malicious Web Servers. Retrieved Oct 31, 2009 from <http://honeynet.org/papers/wek/>

[10]. Rajab, M., Zarfoss, J., Monrose, F., & Terzis, A. (October 2006).

[11]. Multifaceted approach to understanding the botnet phenomenon. Retrieved October 31,2009from http://www.imconf.net/imc 2006/papers/p4-rajab.pdf>

CONFERENCE PAPER
"A National Level Conference on Recent Trends in Information Technology and Technical Symposium" On 09th March 2013
Organized by
Dept. of IT, Jawaharlal Darda Inst. Of Eng. & Tech., Yavatmal (MS), India