July 2016

SURVEY OF BOTNETS AND DETECTION TOOLS

VIVEK ARYA

Guru Gobind Singh Indraprastha University, Sector 16 – C, Dwarka, New Delhi, India.,
88vivekarya@gmail.com

ASHISH CHAUHAN

Guru Gobind Singh Indraprastha University, Sector 16 – C, Dwarka, New Delhi, India.,
ashish_1787@yahoo.co.in

Follow this and additional works at: <https://www.interscience.in/ijcct>

Recommended Citation

ARYA, VIVEK and CHAUHAN, ASHISH (2016) "SURVEY OF BOTNETS AND DETECTION TOOLS,"
International Journal of Computer and Communication Technology. Vol. 7 : Iss. 3 , Article 13.

DOI: 10.47893/IJCCT.2016.1369

Available at: <https://www.interscience.in/ijcct/vol7/iss3/13>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Computer and Communication Technology by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

SURVEY OF BOTNETS AND DETECTION TOOLS

VIVEK ARYA¹ & ASHISH CHAUHAN²

^{1,2}Guru Gobind Singh Indraprastha University, Sector 16 – C, Dwarka, New Delhi, India.
E-mail : 88vivekarya@gmail.com & ashish_1787@yahoo.co.in

Abstract - Botnet meaning bot network is a collection of infected computers under the command and control of a single individual known as a botmaster. Botnets are a current threat to the Internet community and have recently been a root cause of many Internet attacks. Various attacks that include spamming, distributed denial of service attacks, phishing, click fraud, hosting illegal material, key logging etc. are being carried out by hackers using botnets. Botnets are becoming more efficient and more elaborate with time and their use is growing at an exponential rate. In this paper a detailed study of botnet, their topologies, rallying mechanisms and communication protocols used and detection mechanisms both at the network and host level are presented.

Keywords- botnet; bots; command and control server;

I. INTRODUCTION

Botnets constitute one of the biggest current threats to the Internet as well as to individual computers and they are growing at an alarming rate. A botnet is defined as a collection of infected computers called bots under the control of a common master called as the botmaster or the bot herder [1]. The infected computers install and run a remote controlled bot application (a bot is defined as a set of small scripts used to perform automated functions) which the botmaster uses to control these infected machines. A botnet recruits its army by exploiting software vulnerabilities or using social engineering techniques.

The computing power provided by the support of thousands of bots within a botnet make them suitable for carrying out cyber-attacks such as distribution of spam emails, coordination of distributed denial of service attacks, and automated identity theft, e.g. credit card information and general banking data for financial fraud [2].

A. Characterization of Malware

Malware short for malicious software is software designed to disrupt computer information, gather sensitive information, or gain unauthorized access to computer systems. In recent years the diversity of malware has grown exponentially, new variants are coming every day, with constant improvement to the techniques employed and a growing degree of sophistication [3].

Malware can be classified into various categories relating to functions performed by a particular type of a malware. They can be categorized as:

a) **Computer Virus**: A computer virus is a type of malware that propagates by inserting a copy of itself into and becoming a part of another program. It spreads from one computer to other leaving infections as it travels. Viruses spread

upon execution of an executable file to which the virus has been binded onto. Therefore a virus may not spread until the executable has been executed. Virus spread when the software or file they are attached to is transferred from one computer to another through any of the file sharing techniques [4].

- b) **Computer Worms**: A computer worm uses the network to send copies of themselves to other PCs. Worms are standalone softwares and unlike viruses do not need any host program to spread and propagate. A worm enters a computer through vulnerability in the system and takes advantage of file-transport or information-transport features on the system to allow it to travel unaided.
- c) **Computer Trojans**: A Trojan is a harmful piece of software that appears to look legitimate. Users are tricked into downloading and installing a Trojan which appears to be software fulfilling a certain purpose. On activation, it can achieve a number of tasks on the host machine such as damaging or stealing of data, displaying annoying ads or popups on the user machine on a continuous basis. Trojans are also known to create back doors to give malicious users access to the system. Unlike viruses and worms, Trojans do not reproduce by infecting files, neither they self-replicate. Trojans spread by downloading and installing rogue and fake softwares [5].
- d) **Computer Bots**: Bot is derived from the word robot that signifies an automated processing of tasks. A typical use of bot is to gather information (e.g. web crawlers), or interact with Instant Messaging (IM), Internet Relay Chat (IRC), or other web interfaces. Bots can also be used for either good or malicious intent. A malicious bot is a piece of malware that is designed to infect an host and connect back to a command and control server. The command and

control server manages the entire network of compromised hosts (infected hosts) and this network is commonly called as the Botnet. The main difference between a bot agent and a common piece of malware lies within a bots ability to communicate with a command and control (C&C) infrastructure. Using the command and control infrastructure the bot is able to receive commands and instructions from the botmaster. This compromised host can now be used for various activities involved with Internet crime.

B. A Brief History and Impact of Botnets

Historically, the first bots were programs used in Internet Relay Chat (IRC) networks. Internet Relay Chat was developed in the late 1980's and allows users to talk to each other via IRC channels in real time. Bots offered services to other users – for e.g. greeting notifications when a new user enters a specific channel or a chat room, a simple word guessing game etc. The first bot program EggDrop created by Jeff Fisher in 1993 originated as a useful feature in Internet Relay chat for text based conferencing on many machines in a distributed fashion [6].

The first known malicious bot was discovered in 1999, known as the Pretty Park worm. It contained a limited set of functionality and features, such as the ability to connect to a remote IRC server, retrieve basic system information, login names, email addresses and nicknames [7]. It had the ability to be remotely controlled through an IRC channel. In 1998, emerged the GT-Bots (Global Threat Bots). This is a common name used for all mIRC- scripted bots. Their remote control mechanism was IRC and these bots launched an instance of the mIRC chat client with a set of scripts and other binaries. They used the HideWindow executable to make the mIRC instance unseen. Their main drawback was their large size which was greater than 1 MB.

Then in 2002 came the most powerful and notorious of all Agobot/Gaobot and its variants Phatbot, Forbot, Xtrmbot etc [8]. Still out in the open these bots were assumed to be the best known family of bots. The bot with a professional design was written in C++ with cross platform capabilities and the source code put under the GPL. For its command and control it used a IRC based C&C server. These bots were used for carrying out Denial of Service attacks and for retrieving sensitive information from the victim. Along with Agobot came the most active bots in the wild currently, the SDBot and its variants (Rbot, Urbot, UrXBot, SpyBot). These bots were written in C and thousands of versions currently exist. It offers similar features as AgoBot and is open source bot, although the command set is not as large nor the implementation as sophisticated as Agobot. Some variants of SDBot apart from using IRC as

command and control server also used HTTP to command the bots.

Various new and strong botnets have been created till date such as Storm, Srizbi, Cutwail, Ghag, Mega-D, Asprox, Rustock, Warezor, Doubot, Ozdok, Xarvester, Koobface, Zeus, Waledac, SpyEye, Conficker, Cythosia etc. A typical botnet may comprise of thousands of Zombie hosts and thus present a tremendous threat to the Internet security and privacy. Nowadays bots are being designed by criminal elements or even criminal organizations to achieve more specific targets. The types of attacks carried out by botnets include [9]:

- a) Distributed Denial-of-Service Attacks (DDOS attacks): A DDOS attack is an attack on a computer system or network that causes loss of service to the users, typically the loss of network connectivity and services by consuming the bandwidth of the victim's network or overloading the computational resources of the victim's network. The term distributed defines the presence of multiple sources being used in carrying out the attack. DDOS attacks are not only limited to web servers, virtually any service available on the internet can be a target of such type of an attack.
- b) Spamming: Emails coming from untrusted or vague email addresses consisting of advertisements, false information relating to demand of money, viruses, malware, phishing mails etc are called as spam. Some bots offer the possibility to open a SOCKS v4/v5 proxy (a generic proxy protocol for TCP/IP based networking applications) on a compromised machine. The SOCKS proxy's main purpose is to hide the attackers true IP address. After enabling the SOCKS proxy, the infected machine can then be used for spamming. With the help of thousands of bots in a botnet, an attacker is able to send massive amounts of bulk emails.
- c) Keylogging: The keylogging technique is used by various bots in order to steal sensitive information. Keylogging technique works best when the communication is carried over a secure channel (e.g. SSL or VPN) because data is keylogged prior to encryption. A Keylogger is a piece of software that records everything that is typed. Various banking bots use the keylogging technique to gather various banking credentials such as credit card numbers, pin and tan nos.
- d) Click Fraud: It is a type of money earning scheme in which a bot visits a website pretending to be genuine users and clicks on advertisements purporting to make profit for a website owner. This fraud takes money from online advertisement companies who pay a small amount per click. The Google AdSense abuse is a famous example of such kind of fraud. It is

very hard to detect as every click supposedly comes from a different IP address.

- e) Warez and Malware: Illegal material such as pirated software, pirated games, e-books can be stored as a dynamic repository on an infected host. This material may often contain malicious code in the form of viruses, Trojans, spyware and adware. The bot is also capable of stealing serial numbers and licenses for genuine software installed on the host's machine. The bot masters can later sell these licenses at cheaper rate in the black market.

C. Botnet Taxonomy

Botnet Taxonomy is purely classified on the basis of the location of the command and control (C&C) server. The command and control server functions as the brain of the bot network. The command and control server is controlled by the botmaster or the bot herder and is used to send commands and instructions to the bots. The bots connect to the command and control server on a regular timely basis. It can be considered as a master controlling its slaves. The bots function according to the commands they receive from the C&C server [10]. Botnet topologies are rapidly evolving in order to escape shutdowns, hijacking attempts and commercial defenses.

1) Star or Centralized Command and Control Architecture

This topology considers a single command and control server to which all the infected hosts (bots) are connected [5]. All the bots communicate with this C&C server and the server is able to communicate with all the online bots simultaneously.

Figure 1. Star based Command and Control Architecture

The strength of a star based C&C topology is that instructions can be transferred rapidly as there is direct communication between the C&C and bot agents. However it suffers a serious drawback of single point of failure.

2) Multi Server Command and Control Architecture

The multi-server command and control architecture is a logical extension of the star topology. It contains multiple servers as C&C servers to overcome the single point of failure problem. These multiple servers communicate with each other while managing the botnet. If a particular server is traced out and rendered dysfunctional the remaining servers continue to control the botnet.

Using a multi-server based approach the botmaster can allocate the bots to servers in their common geographical area. This speeds up the communication between the bots and the servers. However creating an empire of bots using the multi-server approach takes significant planning and effort.

Figure 2. Multi Server Command and Control Architecture

3) Hierarchical Command and Control Architecture

In Hierarchical based approach bots have the ability to proxy new C&C instructions to previously propagated bots. However updated command instructions typically suffer latency issues making it difficult for a botmaster to use the botnet for real time activities [6].

In a hierarchical botnet no single bot is aware of the location of the entire botnet. This configuration makes it difficult for security analysts to determine the entire size of the working botnet.

Figure 3. Hierarchical Command and Control Architecture

4) *Random or P2P (peer to peer) Command and Control Architecture*

The random or peer to peer based C&C architecture can also be considered as a decentralized architecture. This architecture is very difficult to locate. The instructions are given to the bot network by injecting the instructions to any one of the bot agent [7]. The injected bot then participates in communication with other bots it connects to and injects the instructions into them. This process continues repeatedly until every bot in the network receives the instructions provided by the botmaster.

Figure 4. Random or P2P Command and Control Architecture

D. *Rallying Mechanisms and Communication Protocols*

The rallying mechanisms define how the bot is able to locate the C&C server(s) in order to receive commands and instructions. Three main methods exist:

- Hard Coded IP addresses of C&C servers are embedded in the bots binary file.
- Dynamic DNS Domain Name: A bot master buys a fully qualified domain name (FQDN) or uses one provided by dynamic DNS providers. Attackers often configure the bot to connect to a fully qualified domain name rather than a hard coded IP address. The disadvantage of using a hard coded IP address is that if the IP address is detected, it can be blocked using a firewall or it can be removed from the internet (routed to a black hole for physically shutdown), the attacker would lose total control over the botnet. If an FQDN is used, both the IP address and the

FQDN would have to be removed before the attacker would lose total control over the botnet.

- Distributed DNS Service: In this mechanism the bot master runs its own DNS server which resolves the IP addresses of the C&C servers. Bots have the IP address of the DNS server hard coded in the binary files. This method is the most difficult to detect.

The communication protocols used in a botnet describes how the bots and the command and control server talk to each other. The communication protocol is used to send instructions and commands from the command and control server to the bots. The bots also send stolen data to the server using the protocol defined. Traditionally the communication pattern being used was via Internet Relay Chat (IRC). The bot would join a private IRC server, normally protected with a password. The bot received instructions from the IRC channel in the form of title messages of the IRC channel.

Recent bots use the Hypertext Transfer Protocol (HTTP) because many firewalls especially corporate networks, allow HTTP traffic. An example of a botnet that uses the HTTP protocol for communication is Zeus. Some of the latest bots are adopting themselves towards p2p file sharing protocols.

II. ANALYSIS AND DETECTION OF BOTNETS

In order to understand and analyze the working of a botnet, a suitable sandboxed environment is required. Then various software tools and techniques can be used to gather evidence of a botnet infection. Various detection methods have been suggested in detecting the presence of bot activity [14]:

A. *Network Based Detection*

Lot of tools and techniques exist for analyzing network traffic to identify normal behavior. Network monitoring tools like Wireshark etc. can be used for observing traffic going in and out of a computer. Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS) and firewalls are used to identify, detect or prevent malicious network traffic from harming computer systems. Regarding network traffic of botnets, this can be hard to identify and separate from normal behavior, since they more often apply advanced techniques, e.g. data encryption and use common network protocols used for network activity.

Bots always connect to their C&C server (except p2p based botnets) in order to be controlled by the botmaster. Since many botnets use IRC C&C servers it becomes very easy to detect such type of botnets. Normally IRC traffic flows to and fro on ports 6667,

6660-69 and 7000. As IRC is not a widely used protocol among normal users these days so monitoring traffic on these ports can be used in identifying bot infection.

Phoning home mechanism a prerequisite for all botnets generates network traffic, hence traffic to known botnet C&C's can be checked by IDS (using known or customized IDS signature). Other suspicious network traffic, like DNS requests against a specific DNS server or huge amount of SMTP or FTP, TFTP traffic coming from several machines on a network should raise an alarm.

For detecting abnormal incoming or outgoing network traffic, an IDS (e.g. SNORT or Bro) and advanced firewalls (Intrusion Prevention Systems) are used. These are placed between the internet and the local network (referred to as a honeynet). This is used only used to monitor and log the network activity. The logs can then be used in tracing the botnet connections, revealing and storing evidence of malicious action's existence and origin.

There are a few applications that currently exist for the detection of botnets. Few of them listed are:

1. **BotHunter:** It is a type of NIDS that uses network dialog correlation algorithms (behavior based detection) and was developed by Cyber-TA research program in order to identify infected machines. It uses a snort engine to monitor communication between the internet and internal network. The system also has an n anonymous repository reporting system that adds newly discovered threats based on the information gleaned from the users, which increases its accuracy and ability to detect new botnet threats. Bothunter is also capable of detecting p2p botnets [15].
2. **Rishi:** Rishi is a tool for identifying infected hosts that are controlled as bots[16]. This is done by passively monitoring network activity. This tool acts as an NIDS and is written in python, it monitors captured TCP packets which reveal specific IRC commands of infected hosts. This tool can only detect botnets which use IRC servers.
3. **Snort:** Snort is an NIDS, which was first released by Martin Roesch in 1988. Due to its source code available for the general public it is currently the mostly widely used NIDS. Snort uses a knowledge based detection, though it was not specifically designed to detect botnets. Snort allows users to define their custom based rules in order to analyze network traffic and detect presence of botnets. However Snort is designed for advanced users only as it has a very complex installation procedure on both the windows and UNIX based operating systems. Due to this limitation end users cannot use Snort in its current form [17].
4. **iSensor:** iSensor was developed by SecureWorks and is used on their client networks. It

was originally based on Snort but unlike Snort is not open source [18]. The techniques used by iSensor in detecting botnet activity are signature deployment, anomaly detection, protocol recognition, behavior based, heuristics human analysis of patterns. It also provides packet filtering firewall to block malicious traffic. It provides good detection and protection however is not cheap and easy to setup.

B. Host Based Detection

Host based detection of botnets is done at the host itself rather than observing traffic flow patterns on the network. The malicious code of botnets on analysis has been observed to follow certain patterns and behaviors on the infected host machine. The bot binary file and the executable can be studied to extract information about the bot's activity. A sandboxed environment must be created to execute the bot. After analysis appropriate signatures can be created for botnet detection. Various identifiable patterns have been observed they are:

- a. **Modification of the System Folder:** Most bots place an executable copy of itself in the native system folder of the infected computer. This folder is usually C:\Windows\System32. The file placed in the system folder is usually given a typical system file name and is assigned read only access rights, and is marked as a hidden file. The date of the file and other timestamps are copied from some other typical windows system file so that the file appears to be a legit system file created at the time when the operating system was first installed.
- b. **Modification of the Registry:** Most bots modify the registry values to gain administrative rights. A bot may modify the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon userinit registry value to start itself on every startup. A bot may disable the firewall by modifying the registry values corresponding to the firewall settings. It can enable proxy settings, generate unique ids. Some bots may disable registry settings of services running on the host. Some bots disable the installation of Windows XP SP2, Microsoft updates etc.
- c. **Hosts and other files:** The hosts file used for mapping hosts to IP-addresses locally on a computer, sometimes has values directed to antivirus and security update websites. The hosts file is located in the %System%\drivers\etc folder under the name of hosts.ini. The hosts.ini file can be modified to prevent the system or user from accessing antivirus and security web sites. Bots also create new additional files in the infected system. These files can be used to store the downloaded configuration files and to store temporarily the keylogged data which is then send to the command and control server after specific time intervals.

d. Keylogging and Data Capturing: Keylogging and data capturing are other identifiable patterns on infected hosts. These spyware mechanisms can be used to obtain personal and sensitive information for identity thefts e.g. usernames and passwords, personal identification numbers (pin nos), transaction authentication numbers (tan nos), credit card numbers etc. Since the logged key strokes and captured data also need to be sent out of the host to the botmaster, it creates abnormal traffic that can be identified. However recently more advanced keyloggers are being embedded in the bot functionality. These keyloggers are triggered when certain web sites of banks and web shops are visited by the machines. This makes them very target specific and harder to detect by security analysts, because of its low interactions with the system.

e. Along with manual detection of known bot files and registry changes, software and tools can be used for this procedure to be automated. Standard signature based anti-viruses can only detect known malware which already has a defined signature. When it comes to sophisticated malwares the signature update of malware might be spread after the system is infected, which could mean it is too late. The malware might have already disabled the antivirus, firewall etc.

CONCLUSION

The taking down and shutting a botnet has become near to impossible with the rapid advances being made in botnet infrastructure and distribution. The bot masters are now moving to the clouds to host their botnet. Bot masters could leverage the power of the cloud to create virtual computers and virtual networks containing their bots. The Amazon Elastic Cloud is an example of a cloud which is available to the public and can be used to create and run virtual machines at a very small cost. It allows upto 100 virtual computers per account and a botmaster could create several accounts, possibly using stolen credit cards. Once the bots are setup the bot master can carry out activities such DDOS, spamming, click fraud and then deleting the account making them very hard to discover.

REFERENCES

- [1] Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis, (2006), "A Multifaceted Approach to Understanding the Botnet Phenomenon". <http://conferences.sigcomm.org/imc/2006/papers/p4-rajab.pdf>
- [2] Maryam Feily, Alireza Shahreshtani, and Sureswaran Ramadass, (2009), "A Survey of Botnet and Botnet Detection", in Third Conference of Emerging Security Information, Systems and Technologies. <http://dl.acm.org/citation.cfm?id=1607877>
- [3] See: en.wikipedia.org/wiki/Malware
- [4] "What is the difference between viruses, worms, Trojans, bots?" www.cisco.com/web/about/security/intelligence/virus-worms-diffs.html
- [5] www.sans.org/security-resources/glossary-of-terms
- [6] Banday, M.T, Qadri, J.A., Shah, N.A, (2009), "Study of Botnets and Their Threats to Internet Security," Sprouts: Working Papers on Information Systems, 9(24). <http://sprouts.aisnet.org/9-24>
- [7] Puri R. (2003). "Bots and Botnet: An Overview", SANS Institute. <http://www.sans.org/rr/whitepapers/malicious/1299.php>
- [8] "Virtual Honeypots: From Botnet Tracking to Intrusion Detection" by Niels Provos, Thorsten Holz. See: Chapter 11 – Tracking Botnets. Publisher: Addison Wesley Professional, Publishing Date: July 16, 2007, Print ISBN No 10: 0-321-33632-1. http://ptgmedia.pearsoncmg.com/images/9780321336323/samplechapter/0321336321_CH11.pdf
- [9] Nicholas Janelli, Aaron Hackworth, (2005), "Botnets as a Vehicle for Online Crime", CERT Coordination Centre. www.cert.org/archive/pdf/Botnets.pdf
- [10] Gunter Ollmann (VP of Research, Damballa Inc), "Bonet Communication Topologies - Understanding the Intricacies of Botnet Command and Control". [http://www.damballa.com/downloads/r_pubs/WP%20Botnet%20Communications%20Primer%20\(2009-06-04\).pdf](http://www.damballa.com/downloads/r_pubs/WP%20Botnet%20Communications%20Primer%20(2009-06-04).pdf)
- [11] Daniel Plohmann, Elmer Gerhards – Padilla, Felix Leder, "Botnets: Detection, Measurement, Disinfection and Defence", ENISA. <http://www.enisa.europa.eu/activities/Resilience-and-CIIP/networks-and-services-resilience/botnets/botnets-measurement-detection-disinfection-and-defence>
- [12] B Stiller et Al. (Eds.): Communication Systems IV. Technical Report No IFI 2011.0001, Jan' 2011, Communication System Group (CSG), Department of Informatics, University of Zurich. http://www.ifi.uzh.ch/pax/uploads/pdf/publication/1620/T_R2011_01_0.pdf
- [13] Ping Wang, Shern Sparks, Cliff C. Zou, (2010), "An Advanced Hybrid Peer to Peer Botnet". http://static.usenix.org/event/hotbots07/tech/full_papers/wang/wang.pdf
- [14] Anders Orsten Flaglien (Gjovik University College), "Computational Forensics for Botnet Analyzation and Botnet Attack Mitigation". http://www.andersof.net/wp-content/uploads/CF_paper-Botnets-Anders_O_Flaglien.pdf
- [15] Guofei Gu, Phillip Porras, Vinod Yegneswaran, Martin Fong, Wenke Lee, "Bothunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation." In Proceedings of 16th USENIX Security Symposium, 2007 <http://www.cyber-ta.org/pubs/botHunter-final7.pdf>
- [16] Jan Goebel, Thorsten Holz, "Rishi: Identify Bot Contaminated Hosts by IRC Nickname Evaluation." In Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets, 2007 http://static.usenix.org/event/hotbots07/tech/full_papers/goebel/goebel.pdf
- [17] M. Roesch, "Snort – Lightweight Intrusion Detection for Networks." In Proceedings of USENIX LISA'99, 1999. http://static.usenix.org/event/lisa99/full_papers/roesch/roesch.pdf
- [18] Dell: ISensor Intrusion Prevention Appliance