RESEARCHERID
THOMSON REUTERS

[Dwivedi * *et al.*, 6(7): July, 2017]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

RECENT TRENDS IN BOTNET RESEARCH

Sanjeev Kumar Dwivedi[*1] **and Ankur Singh Bist**[2]
[*1&2]IT & CSE Department, KIET, Ghaziabad

DOI: 10.5281/zenodo.827496

ABSTRACT
With the advent of internet technology and the increased dependency on the internet to carry out financial transactions gave rise to a new generation of malware called botnets. A botnet collectively termed for a network of infected computers or bots are used to carry out various attacks on the internet community. These attacks ranges from DDOS attacks performed on an organization, spamming campaigns, to sample key logging attacks performed on general individuals. Botnets thus are a network of malware infected machines that are under the control of a single or a group of individuals called as the botmasters or botherders. The botmasters sends commands to these infected networks of bots, to which these bots gleefully respond. Botnets are becoming more elaborate and efficient over time and thus the use of botnets is growing at an exponential rate, threatening the average user and business alike.

I. INTRODUCTION
Overview
Context
A botnet is defined as a collection of infected computers called bots under the control of a common master called as the botherder [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. Botnet pose a significant and growing threat against cyber-security as they provide a distributed platform for many cyber-crimes such as distribution of spam emails, coordination of distributed denial of services attacks, and automated identity theft, e.g. credit card information and general banking for financial fraud [2].

This project aims to research botnets and other related threats in order to develop an application designed to detect the possible activities of a botnet and particularly how it can be detected at early stages and finally towards the successful removal of the bot. In this context, the first aim of this project is to produce a simple botnet and implement it into a virtual machine. Then the second aim will be analyze the interaction made by the botnet within the virtual network and makes statistics about its rates of detection.

Background
In only two decades botnet have gone from simple scripts that demonstrated primitive artificial intelligence and automation to a dangerous and devastating force on the internet. Botnets constitute one of the biggest current threats to the internet as well as to individual computers at an alarming rate. The world's usage on computers is growing at a very alarming rate and there is potential for businesses or individual to lose millions of pounds when they got infected.

The first bot "pretty park" was introduced in 1999 on the Internet Relay Chat (IRC) designed for group communication in discussion forums, known as IRC channels. This bot application had the functionality of many of today's bot clients. It was capable of retrieving system information, email addresses, and user credentials and could redirect traffic, launch DDOS attacks, and complete file transfers [2][3]. Most important, the bot was able to update itself and act as an IRC clients and it allowed an administrator to remotely control a large pool of computers using IRC channels. Another bot application appeared in 1999 that made more progress towards powerful, large-scale botnet. The bot Subseven was marked as a Trojan which had remote administration capabilities. The bot was capable of stealing passwords. Logging keystrokes and giving administrative rights to the botnet operator.

The first widespread use of botnet was seen in 2002 when a Russian programmer created a new bot client in c++ that was capable of exploiting critical vulnerabilities in order to infect victim machines. The bot was named as SDBOT. SDBOT was capable of exploiting vulnerabilities in common windows services including NETBIOS, MS PRC etc. What was most revolutionary about SDBOT was that its creator published its source code on a website. Thus giving rise to a number of SDBOT variants. By 2003, bots were increasingly transforming from a method for attacking networks to a way of collecting data on their victims.

In July 2007, a bot named ZEUS identified as a banking bot was used to steal information from the United States department of transport. Since then ZEUS has evolved over time. ZEUS reached record numbers in May 2009 with more than 5000 variants. ZEUS likely originated in Russia and is one of the most pervasive and damaging banking malware till date. The ZEUS malware alone is estimated to have caused damages worth US$100 million since its inception [4]. In July 2009, a report by Trusteer (a security company) stated that 3.6 millions pc's had been infected by the ZEUS botnet in us alone.

II. TECHNOLOGIES

Introduction

A brief history of botnet along with the possible attacks carried out by botnets on individual and organization is given. The main focal point of the review is a depth-study of botnet taxonomy; communication protocols used and rally mechanisms. Various botnet detection tools and methodologies both at the network level and host level are discussed. The chapter ends with the future of botnets, which describes the advancements and sophistication in botnet in the years to come.

Malware Characterization

Malicious software (malware) exploits vulnerabilities in computing system. Malware includes viruses, worms, Trojan horses, spyware that gather information about a computer user and access to a system without permission. It can appear in the form of code, scripts, active content, or other software. Malware programs are divided into 2 classes: first class of malwares needs a host program (viruses, Trojan horses, logic bombs, trapdoors) and second class of malwares are independent programs (worms, zombie). Other categorization of malwares are: first do not replicated (activated by trigger) and others that producing copies of themselves [24-30].

- **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 spread from one computer to other leaving infections as it travels. Viruses spread upon execution of an executable file to which the virus has been bind 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.

- **Computer Worms :-** A computer worms uses network to send copies of themselves to other PCs. Worms are standalone software 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 advantages of file-transport or information-transport features on the system to allow it to travel unaided.

- **Computer Trojans: -** A Trojan is a harmful piece of software that appears to look legitimate. Users are tricked into downloading and installing a Trojan ehich 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, display 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 software's.
- **Computer Bots: -** Bot is derived from the word "ROBOT" that signifies an automated processing of tasks. Botnet has become the most serious security threats on the current internet infrastructure. Botnet (BotNetwork) is an interconnected collection of compromised bots, which are remotely controlled by

its originator (BotMaster) under a common command-and-control infrastructure. Bot is a new type of malware which is designed for malicious activity. After the bot code has been installed into a computer, the computer becomes a bot. BotMaster (BotHerder) is a person which control the remote bots. A typical use of bot is to gather information (web crawlers), or interact with instant messaging (IM), internet relay chat (IRC), OR other web interfaces.

Malware (especially viruses and worms) are self-replicate programs. Viruses require user interaction and propagate slower than worms because it needs user interaction. But worms do not require user interaction and propagate quickly. Actually both are self-propagating programs. But Bots are different from other types of malware. Here all the bots are under the controlled of BotMaster. So if bot exist in computer, it is not harmful until it receives command from BotMaster. After receiving the command from BotMaster, it is dangerous for system. These bots are not self-propagate from one system/network to other system/network. They are in idle state. After receiving the commands from BotMaster, they propagate from one system/network to system/network and to malicious activities.

History of Botnet

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 notification when a new user enters a specific channel or a chat room, a simple word guessing name etc. the first bot program ENGGDROP created by Jelf Fisher in 1993 originated as a useful feature in Internet Relay Chat for text based conferencing on many machines in a distributed fashion [2][5].

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. 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 names used for all MIRC-scripts 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 1MB [21-23].

Then in 2002 came the most powerful and notorious of all Agobot/Gaobot and its variants Phatbot, Forbot, Xtrmbot etc[6]. 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 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 vriant (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.

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 botherder and is used to send command and instructions to the bots. The bots connect to the command and control server on a regularly timely basis. It can be considered as a master controlling its salves. The bots function according to the commands they receive from the C&C server [2][5][7].
Classification of Botnet

Botnets are classified based on
(1) Command and Control (C&C) based architecture.
(2) Protocols used for communication. (IRC, HTTP etc)

Classification based on command and control (C&C) architecture

(1) Centralized architecture
(2) Decentralized architecture

(1) Centralized Architecture

In centralized architecture Botmaster and Bots are not directly connected to each other. They interacted with the help of server (C&C server). All bots are connected with server. Botmaster sit remotely. Here Botmaster give commands to server. When bots are communicated with server, commands are redirected to this bots. After receiving the commands they do malicious activity. i.e. here all the connections happens through server. So C&C server is critical point in this architecture. The main advantage of this model is small latency (delay) which causes Botmaster easily arranges Botnet and launch attacks. But C&C server is weak point in this architecture. If somebody discovers the C&C server and shutdown the C&C server then entire Botnet becomes useless and ineffective. So if attacker uses this architecture than it is easy to prevent our system by these bots. IRC and HTTP are 2 common protocol that centralized architecture used for communication. So we concentrate on decentralized architecture.

The strength of this architecture lies in the fact that instructions can be transferred rapidly as there is direct communication between the C&C server and bots. This architecture offers a low reaction times and good means of coordination. Direct feedback enables easy monitoring of the botnet status for the botmaster and gives information about fundamental properties such as number of active bots.

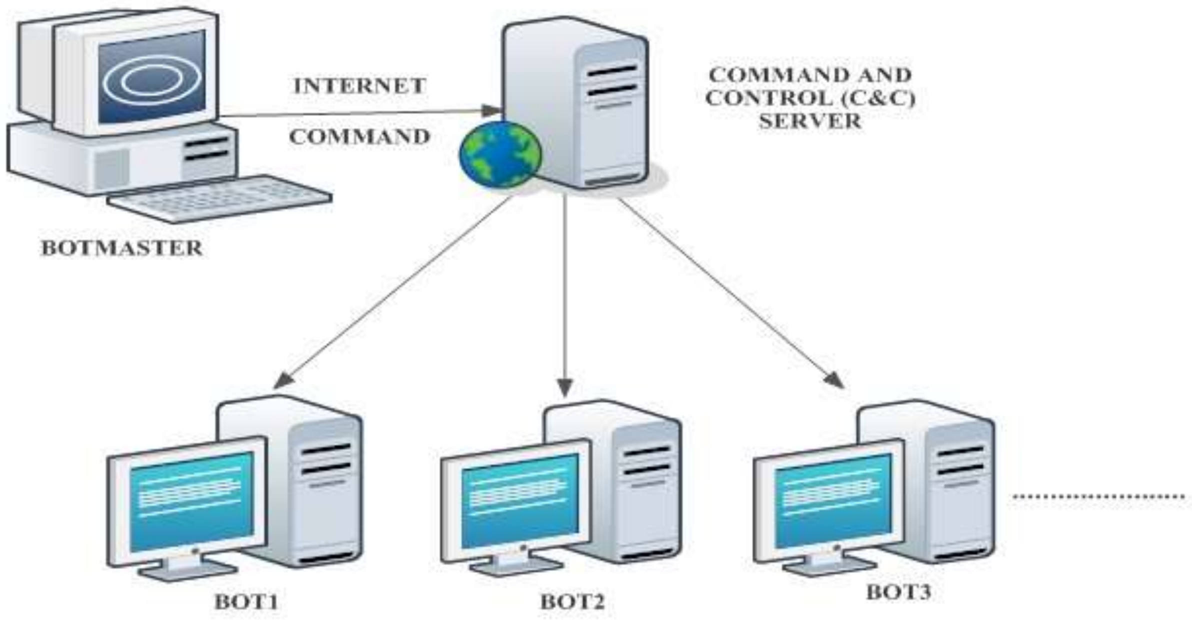

Figure 2.1: Centralized Architecture

(2) Decentralized Architecture

In decentralized architecture there is no central point for communication. Here bots act as a client or server or both. Each bots connected to some other bots of the botnet. So if some bots are offline than botnet still continue to perform his activity. In this architecture botmaster inject commands to ant bot and than this bot broadcast it to all other nodes. Since decentralized botnet allows commands to be injected at any node in the network, so the authentication of commands become essential to prevent other nodes from injecting incorrect commands. Example of decentralized architecture is P2P BOTNET. So botnet that uses P2P based model impose bigger challenge for defense of network [7].

The P2P based botnet are highly resilient to shutdown because they lack of a single centralized command and control server. Multiple communication paths exist between the bots and hence it becomes nearly impossible to detect the root of infection. However the P2P based botnet architecture suffers from latency issues as there is no direct communication between the server and bots. Therefore success rate of instruction and command execution is low. And being P2P nature involves the complexity of management.

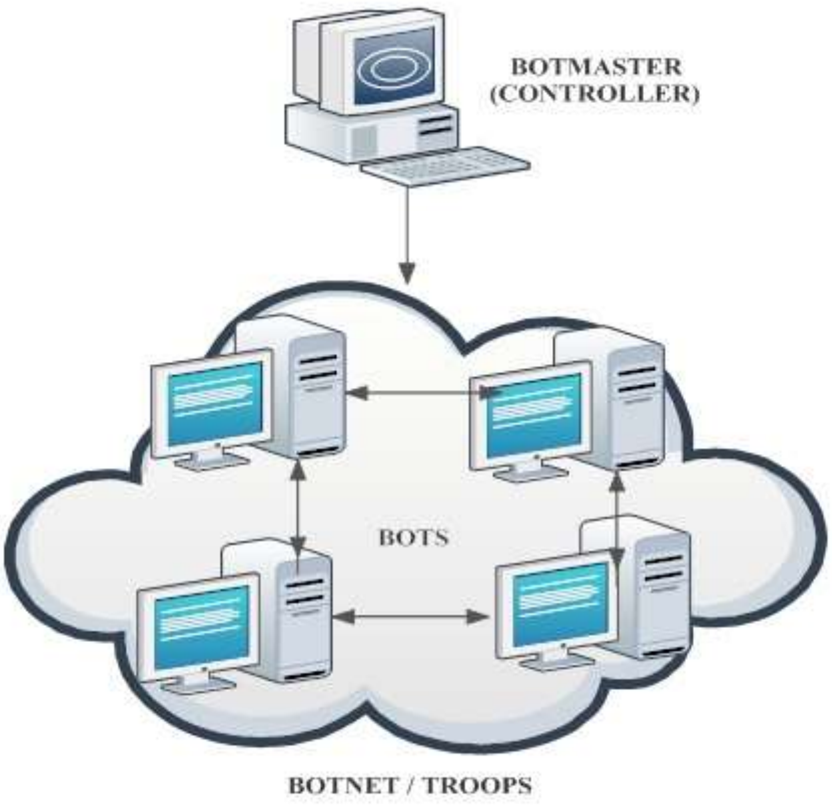

Figure 2.2: Decentralized Architecture

Table 2.1: Comparison of Command and Control Topologies

Topology	**Design Complexity**	**Delectability**	**Message Latency**	**Survivability**
Centralized	Low	Medium	Low	Low
Decentralized	Medium	Low	Medium	Medium
Unstructured	Low	High	High	High

Table 2.2: Types of Bots [8]

Types	**Features**
Agobot Phatbot Forbot Xtrembot	They are so prevalent that over 500 variants exist in the internet today. Agobot is the only bot that can cause other control protocols besides IRC. It offers various approaches to hide bots on the compromised hosts, including NTFS Alternate data stream, Polymorphic Encrypt or engine and antivirus killer.
SDBot RBot UrBot UrXbot	SDBot is the basis of the other three bots and probably many more. 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.
SpyBot NetBIOS Kuang Netdevil KaZaa	There are hundreds of variants of SpyBot nowadays .Most of their C2 frameworks appear to be shared with or evolved from SDBot . But it doesn't provide accountability or conceal their malicious purpose in Codebase.
mIRC-based GT-Bots	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 the user.

DSNX Bots	The DSNX (Data Spy Network X) bot has a convenient plug-in interface for adding a new function. Albeit the default version does not meet the requirement of spreaders, plugins can help to address this problem .
Q8 Bots	It is designed for Unix/Linux OS with the common features of a bot, such as dynamic HTTP updating, various DDOS-attacks, execution of arbitrary commands etc.
Kaiten	It is quite similar to Q8 Bots due to the same runtime environment and lacking of spreader as well. Kaiten has an easy remote shell, thus it is convenient to check further vulnerabilities via IRC.
Perl-based bots	Many variants written on Perl nowadays. They are so small that only have a few hundred lines of the bots code. Thus, limited fundamental commands are available for attacks, especially for DDoS-attacks in Unix-based systems.

Protocol Used For Communication

The idea of botnets originated from Internet Relay Chat (IRC), a text based chat system that organizes communication in channels. The IRC protocol still serves as an important technology for botnet control and enables a centralized communication model. According to the 2010 Symantec Internet Security Threat Report, 31% of botnets used IRC as a communication protocol in 2009. One important property of this protocol is that the number of potential participants within one channel is technically not limited. This allows the connection of many bots in one such channel and the ability to command them in parallel. Additionally private the collection of many bots in one such channel and the ability to command them in parallel. And private conversations are possible on a one-to-one basis. This enables the direct manipulation of single bots. Because IRC based protocol is text-based, it is easy to implement and customize. In the context of botnets, these properties offer a robust, well-established and easy to implement approach to commanding a botnet. IRC channels for botnet control are either hosted on public IRC servers or on servers owned by the botmaster. The bots usually implement only a subset of the IRC instruction set. This is enough to allow the operator to control the bots and reduce functionally to the required minimum. If their own servers are used, arbitrary modifications to the protocol can be made using their own instruction set and encryptions [7]

A well-known standard used throughout the internet is Hypertext Transfer Protocol (HTTP). HTTP is a protocol most commonly used for delivery over the internet. This includes human-readable content like websites and images; also binary data transported in uploads and downloads. Because of these important features, HTTP is available in nearly every network connected to the internet and is rarely filtered. This is especially interesting for botnet operators, because it makes the protocol viable as a command and control protocol. HTTP bots have to periodically issue request to the target C&C server. A typical example of botnets using HTTP for communication are those generated with the commercial ZEUS crime ware toolkit, featuring a tool for the construction of binaries and a graphical user interface situated on the C&C server.

Botnet Communication Channel Architecture

Each Peer Bot will contain list of its next two peer bots and other two non-peer bot information in seed list. The non-peer bot will have only two entries of peer bot information with the condition that they both peer bot will contain the information of each other. The bot master will pass the command to any one of the Peer bot depending upon the diurnal dynamics that particular bot will be selected for the first command passing to the whole botnetwork. After getting the command by the botmaster the peer bot will share this command to its next neighbor peer bot as will connected non-peer bot that will ensure the effective communication, for the purpose of command passing priority will be given to the peer bot. Because peer bot can work as client as well as server too, and connected to other peer bots. On the basis of this topology the communication will be handled [9].

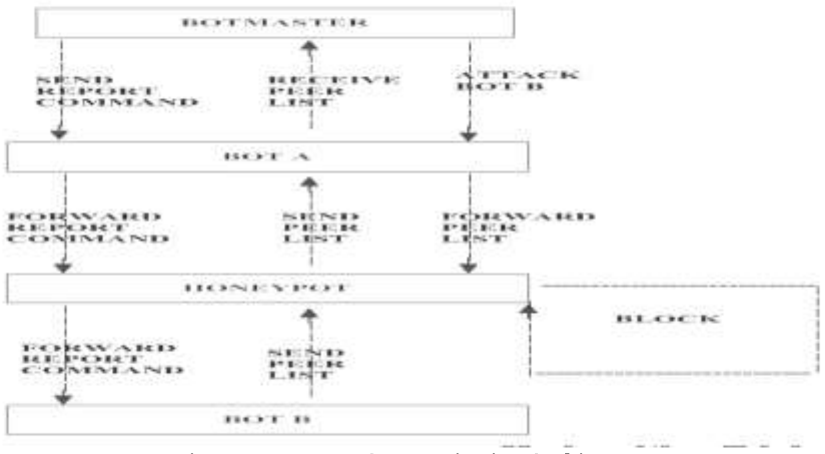

Figure 2.3: Botnet Communication Architecture

Rallying Mechanism
Different rallying mechanism are used by botnet to rally new bots to the C&C server in order to retrieve message from the botmaster. It defines how the infected bots are able to locate the command and control servers [10]. The main methods includes hard-coded IP addresses, dynamic DNS domain name, and distributed DNS services.

(1) Hard-coded IP
Some bots can communicate with the command and control server using a hard-coded IP which means that the IP address of the C&C server is hard-coded into the binary of the botnet**[**11]. This technique is not very popular because once the bot is trapped and has been analyzed, the address of the C&C server can be easily found and shutdown. Once the C&C server is disabled, bots belonging to this server become useless because they cannot receive any more orders from the botmaster.

(2) Dynamic DNS Domain Name
The Dynamic DNS Domain Name uses a hard-coded domain name assigned by dynamic DNS provider, which allows the botmaster to relocate the botnet easily. If the connection fails, the bot sends a DNS query to receive the new domain name of the C&C server. So on the bot does not depend anymore from the C&C server, this one can go down and a new C&C server will be assigned to the bot. A few websites give this free service such as www.dyndns.com. We can create our own domain www.yourname.dyndns.com and assign a dynamic IP to this name. This technique has become more widely used because authorities can often shutdown botnet C&C server. With this method the bot master can set up a new server and update DNS to point to the new server.

Domain Flux can also be used by this method. Domain flux is a technique used by many botnets in order to avoid detection; they do this by changing the point of contact. It generates a list of domain names commonly using Domain Generation Algorithm (DGA) that it will contact depending on the date and time. This is very effective because it makes it very difficult to shutdown or blocks the C&C server because the point of contact changes on a regular basis.

(3) Distributed DNS Services
Another method is to use a Distributed DNS Services. This service is the most sophisticated and is being implemented in the newer generation of botnet. Botnet run their own DNS server in specific locations where authorities cannot reach them (because the law in these places is not up-to-date in the internet security domain). The bots have the IP address of the DNS server hard coded in the binary files. The bots contact its own DNS server to retrieve the IP of the C&C server. These DNS servers use a high port to communicate which makes them difficult to be detected by security devices. They are actually hardest to detect and destroy.

Botnet Attacks and Profit

Botnet can be used for several purposes and they are said to be the major cause of today's internet problems. Botnet consisting of hundred thousands of infected hosts carry enormous bandwidths, sensitive information and computer power [10].The various kinds of activities carried out by the botnet are:-

(1) Distributed Denial of Services
(2) Spamming
(3) Key logging
(4) Click Fraud
(5) Identity Fraud
(6) Warez and malware
(7) Retail

(1) Distributed Denial of Services

A DDOS attack is an attack on a computer system or network that cause 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 of 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.

(2) 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 main purpose is to hide the attackers true IP addresses. After enabling the SOCKS proxy, the infected machine can then be used for spamming. With the help of bots in a botnet, an attacker is able to send massive amount of bulk emails.

(3) Keylogging

The keylogging technique is used by various bots in order to steal sensitive information. Keylogging technique works when the communication is carried over a secure channel (e.g. SSL or VPN) because data is keylogging prior to encryption. A keylogging 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.

(4) Click Fraud

It is a type of money earning scheme in which a bot visits a website pretending to be genuine users and click on advertisement 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.

(5) Identity Fraud

This technique is also called Phishing. Bots can host multiple fake websites pretending to be eBay, PayPal, or a certain bank and harvest personal information. Usually the name change creates subtle confusion (such as parpal.com to epaypl.com will redirect you to a fake paypal.com but with exactly the same contents and looks as the original site.

(6) Warez and Malware

Illegal material such as pirated software, pirated games, e-books can be stored as a dynamic repository on an infected machine. 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 license at cheaper rates in the black market.

(7) Retail

Botnets are now being sold to criminals for large sums. The botmaster who has built an empire of bots by successful social engineering techniques sell his bots to other individuals or organizations. The buyer receives the ability to control part of the botnet in order to perform tasks such as the one mentioned above. An example of this is the Zeus botnet which is sold to criminals as a toolkit at prices ranging from $3000 to $4000.

Behavior of a Botnet

Figure 2.4 depicts the behavior of a botnet. A botmaster correlates the activities of all bots in the botnet through C&C server. After the bots receive the master's command, they immediately respond and simultaneously conduct malicious activities at a set time [11].

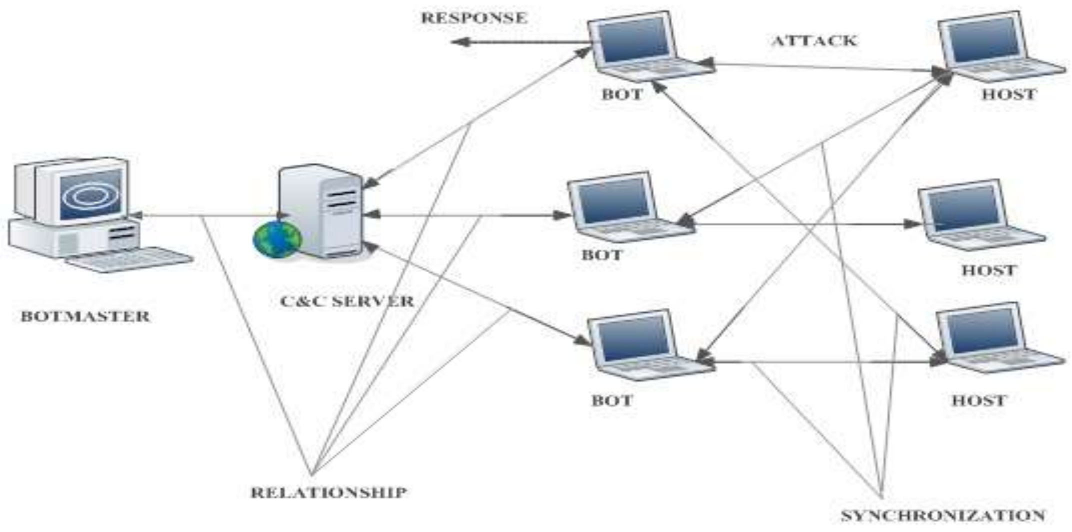

Figure 2.4: Behavior of Botnet: Relationship, Response, And Synchronization

Fig 2.5 shows a comparison of response time between humans and bots. Regarding human behavior, when a legitimate host receives a message, it responds or performs an action from a wide variety of possibilities after a variable thinking time. On the other hand, when a bot receives command from its master, it performs preprogrammed activities with a constant response time [11][12].

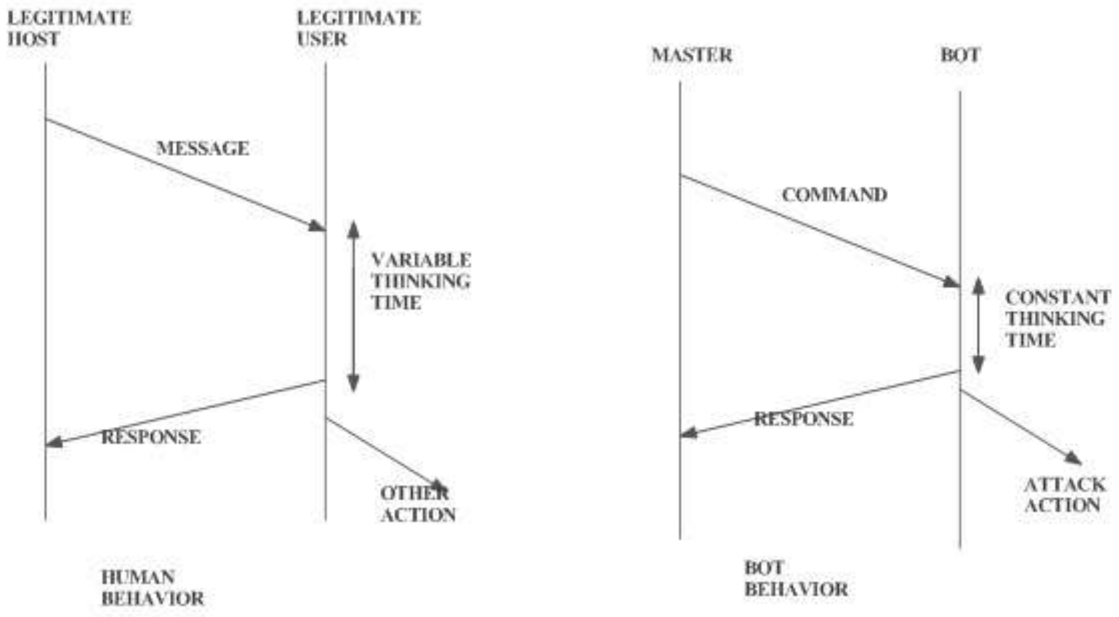

Figure 2.5: Comparison of Response Time Between Humans And Bots

Analysis and Detection Methods

In order to understand and analyze the working of a botnet, a suitable sandboxed environment is required. Then various software tools and technique can be used to gather evidence of a botnet infection. Various detection methods have been suggested in detecting the presence of bot activity [2][12][13].

(A) Network Based Detection
(B) Host Based Detection

(A) Network Based Detection

Lot of tools and technique exist for analyzing network traffic to identity 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 technique, e.g. data encryption and use common network protocols used for network activity [1].

Bots always connect to their C&C server (except P2P based botnet) in order to be controlled by the botmaster. Since many botnets use IRC and C&C servers it becomes very easy to detect such type of botnets. Normally IRC traffic flows to and from 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 [12].

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 arise an alarm.

For detecting abnormal incoming or outgoing network traffic, an IDS (e.g. SNORT or Bro) and advanced firewalls are used. These are placed between the internet and local network (referred to as a honeynet). This is used only 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 botnet. Few of them listed are:-

(1) BotHunter

It is type of NIDS that uses network dialog correlation algorithm (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 anonymous repository system that adds newly discovered threats based on the information glened from the users, which increases its accuracy and ability to detect new botnet threats. Bothunter is also capable of detecting P2P botnets.

(2) Rishi

Rishi is a tool for identify infected hosts that are controlled as bots. This is done by passively monitoring network activity. In order for this tool to work it has to be connected to a centralized router or switch. 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 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.

(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. 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 botnet is done at the host itself rather than observing traffic flow patterns of the network. The malicious code of botnets on analysis has been observed to follow certain patterns and behaviors on the infected host machines [14]. The bot binary file and the executable can be suited to exact 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:-

(1) Modification of the System Folder

Most bots place an executable copy of itself in the native system folder of the infected computer. This folder id 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.

(2) 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 \current version\winlogon userinit registry values to start itself on every startup. A bot may disable the firewall by modifying the registry values corresponding to the firewall setting. It can enable proxt setting, generate unique ids. Some bots may disable registry setting of services running on the hosts. Some bots disable the installation of windows XP SP2, Microsoft updates etc.

(3) Hosts and Other Files:-

The hosts file use 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%\dirves\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 systems. 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.

(4) Network Traffic

Messages and commands have to be transferred over network channels for to bots to be controlled by its botmasters. This creates unexpected traffic not associated with normal network applications on the hosts. Various bot malware open different network ports in order to obtain connection to the C&C server or other bots, and such ports that appear to be open should raise suspicion. When it comes to the network traffic itself, abnormal network activity (e.g. IRC commands, HTTP requests) directed to unknown hosts or IRC servers associated with malware etc. can also be caused by bots. Bots may in some case also create backdoors or connections to their C&C servers which demands attention when analyzing network traffic on possibly infected machines. Establish local HTTP or FTP servers for communication and file sharing/distribution have also occurred on target machines, because of bot infection.

(5) 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 identify thefts e.g. usernames and passwords, personal identification numbers, transaction authentication numbers, 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 bots 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 interaction with the system [13] [14].

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 antivirus can only detect known malware which already has a defined signature. When it comes to sophisticated malwares the signatures 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.

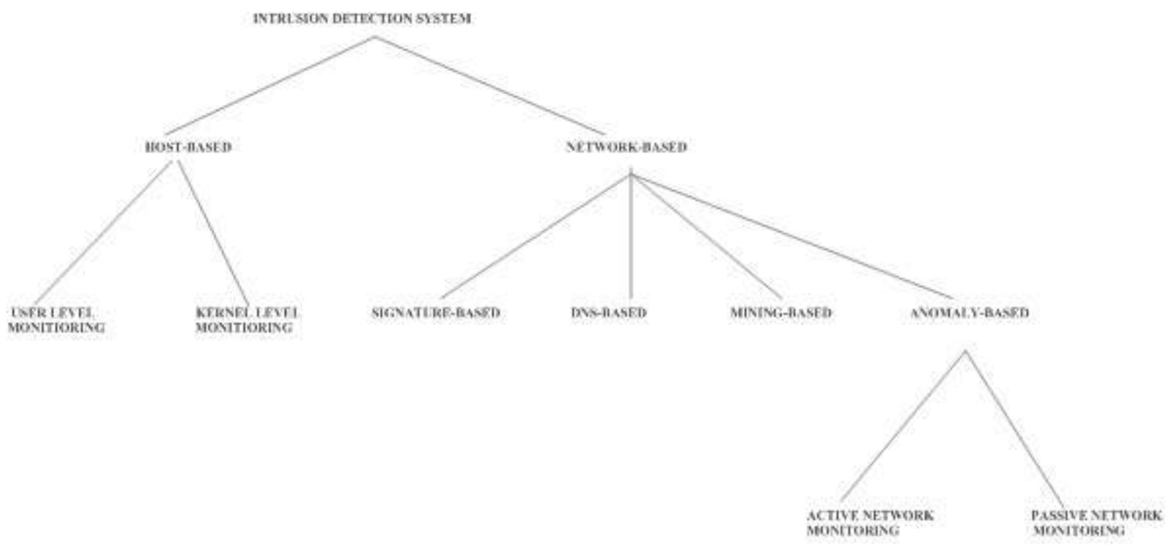

Figure 2.6: Classification of Intrusion Detection System

III. SURVEY OF THE RELATED WORKS

There are many techniques for detection of botnets. There are 2 essential techniques for botnets detection: setting up honey nets and passive monitoring network traffic. Many paper already discussed about using hone nets for botnet detection. We are going to use second method for botnet detection.

Yukiko Sawaya et al. [14] present a flow-based attacker detection method which detect attackers (Nuisance attackers, Simple attackers and Obvious attackers) without predefined blacklist/white list hosts. In their work, they first collect and analyzed traffic flows related to the object port (open port) and decoy port (closed port). For this they generate a weird host list (WHL) for an arbitrary object port. Then they calculate the flow statistics for the obvious attackers to obtain a sample of their tendency which is used for detecting nuisance attackers. The nuisance attackers are identified based on similarity between the feature of each of hosts sending flows to object port and samples (parameter). By using this method they easily detect attackers in a high traffic network without high computing resources. Problem with this method is that for finding nuisance attackers first they find out obvious attackers (generate WHL). Also they used flow statistics (used various parameter) so if they wrongly generate WHL (if miss any closed port) and wrongly measured statistics then finding the nuisance attackers is difficult task. Also they used here flow-based approach so attacks that are injected in payload will not be identified.

Alirena Shahrestani et al. [15] present visualization techniques which enhance the visibility of network traffic related to invariant bot behaviors and provide notification of existence of bots. They developed visual network monitoring tool [(visual threat monitor (VTM)] which is “proactive approach” (triggered before attackers attack) to assist security personnel in bot infection detection in small to medium size network. The visual information provided by VTM will further analyzed by human conceptual ability to gain useful knowledge about botnet activities. The main advantage of this system is that visualized information is easier to be processed and for user it is easy to gain useful knowledge about bot existence in a network. In this approach bot behavior pattern matching will be used as an evidence of botnet existence. The visualization technique used in this system consist of graph, scatter plots and parallel histograms visualization. But these technique will be used to visualized a limited set of invariant bot behavior (like fast response time, small size command etc.). Also based on the metrics they determine whether a node is bot node or not. And these systems allow adjusting different threshold for each parameter to meet certain requirement so if they not fixed threshold properly then finding a bot is difficult task.

Hsiao et al. [16] proposed a method which applied flow correlation for grouping the same activities of same bots and then applies scoring technique to identify normal IRC and abnormal IRC behaviors. In their system input data, network traffic converts into flow data to aggregate the information provided by packets. Each flow has its start and end timestamp, total bytes, and packet transmitted from source to destination. The related flows are correlated according to group activity and homogenous response. The related IRC traffics are applied to differentiate normal IRC and abnormal IRC traffic. But this method is not able to detect unknown bots.

Chunyong Yin et al. [17] identify that botnet uses some common protocol such as IRC, HTTP and P2P for his communication. This makes detection of botnet is a challenging problem especially P2P botnet. Because P2P botnet is a distributed malicious software networks, it is more difficult to detect this bot. Chunyong Yin et al. proposed a P2P botnet detection framework which is based on association between common P2P networks behaviors and host behaviors. This mechanism not only detects known P2P botnet with a high detection rate but also detect some unknown P2P malwares. This method deals with some problems such as data encryption, route selection, communication behaviors.

Table 2.3 Comparisons of Existing Approaches

S.No.	Title	Year	Tool/Algo. / Module	Disadvantage	Advantage
1.	Flow based botnet detection [16]	2009	flow correlation	Attacks which are injected in payload can't be identified.	Able to detect unknown bots.
2.	Deep analysis of intending peer-to-peer botnet [19]	2009	Uses traditional packet inspection for classification of botnet	Didn't provide any approach for detection and measuring of existing botnet or advanced botnet.	Discussion on advanced P2P botnet and present existing botnet.
3.	P2P traffic identification based on the signature of the key packets [23]	2009	Based on the signature of the key packets, flow attributes, signature matching.	Low computational complexity.	Generate false alarm if bot traffic are encrypted.
4.	Discovery of invariant bot behavior through visual network monitoring system [15]	2010	visual network monitoring tool [(visual threat monitor (VTM)]	Visualized a limited set of invariant bot behavior (like fast response time, small size command etc.).	Visualized information is easier to be processed and for user it is easy to gain useful knowledge about bot existence in a network.
5.	Behavioral correlation for detecting P2P bots [18]	2010	Used P2P bots behaviors	Cannot detect protocol independent botnet	Concentrated on P2P botnet detection procedure
6.	An overview of IP flow based intrusion detection [20]	2010	Based on packet sampling, payload based, pattern matching.	It is a logical chance for high speed network.	Absence of payload to be received.
7.	Detection of attackers in service using anomalous host behavior based on traffic flow analysis [14]	2011	Generation of weird host list to find out nuisance attackers	This is flow-based approach so attacks that are injected in payload will not be identified.	Easily detect attackers in a high traffic network without high computing resources.
8.	P2P botnet	2011	Filtering,	filtering the different	Detect both known

	detection based on association between common network behaviors and host behaviors[17]		feature selection of P2P data, host based detection and network based detection	protocols based on ciretria not fixed, deals with certain problem like data encryption, route selection, other communication behaviors.	and unknown P2P malware.

IV. DRAWBACKS OF EXISTING SYSTEM

- How to differentiate between normal communication program and BotNet communication program.
- How many features needed for describing the P2P data properly?
- Attacks that are injected in payload will not be identified.
- Whether detection method discover the unknown P2P botnet. Can't say anything?
- Deal with certain problem such as data encryption, route selection, and communication behaviors.
- Cannot able to analyze bots based on protocol independent and architecture independent.

. .

V. SURVEY SUMMARY

The botnet phenomenon presents several new challenges for internet community. This section defines the classification of different types of malwares, along with the taxonomy of botnets, for better understanding the behavior which is essential to identify and detect the significance rise of botnet activity. This chapter also discussed about the communication protocols used by botnets and the rallying mechanisms being used by the present generation of botnets. By keeping in mind that botnets are moving targets, all aspect from communication protocols, attacks to rallying mechanism are constantly evolving and give a hard task for network defenders.

The constant attacks on the internet community carried out by botnets are also discussed. Current detection tools and technologies for botnet both at the network level and the host level discussed. In the previous work, the existing approaches do not address the problems like data encryption, communication behavior etc. The survey also gives the comparison of existing approaches.

VI. CONCLUSION

The botnet detection is a very challenging research area, especially in the P2P network.The botnet are more concealable and robustness than traditional centralized structured botnet.This P2P botnet detection model based on P2P node detection algorithm, P2P node clustering algorithm, and botnet detection algorithm based on control flow stability, which provides the efficient way to detect the botnet in the network. The main purpose of this report was to delve into the topic of botnets, to gain a deeper understanding on what botnets are, how they work and various detection mechanisms being used to prevent botnet infection.

The first objective of this project was to provide a literature review on botnet history and taxonomy including trends, the attacks carried out by botnets, and also analysis and detection methods currently present for botnet detection and elimination. The literature review produced contained a classification of malware which distinguished between the common variants of malware out in the open, with bots being the current generation of malware. A brief history of botnets was given which traced the past involvement of botnets for conducting internet related crimes. Botnet taxonomy describing all the various architectures seen in botnets along with the various communication protocols and rallying mechanisms used by botnets were discussed. All attacks and profits made by botnets were discussed. Existing detection methods was analyzed, looking for their strengths and weaknesses, and the conclusion attained was that there exists no perfect solution.

VII. REFERENCES

[1] Lei Zhang, Shui Yu, Di Wu, Paul Watters, "A Survey on Latest Botnet Attack and Defense", International Joint Conference Of IEEE Trustcom-11/IEEE ICESS-11/FCST-11, PP.53-60, 2011.
[2] Maryam Feily, Alireza Shahreshtani and Sureswaran Ramadas "A Survey of Botnet and Botnet Detection",IEEE in The Third Conference of Emerging Security Information, Systems and Technologies, pp.79-84, 2009.

[3] Matthew Boyd, “Botnet in Deep Analysis”, Available at:http://iboyd.net/wp-content/uploads/2008/05/ist-451-final-pdf.
[4] Unisys Stealth Solution Team, “ Zeus Malware: Threat Banking Industry”, White Paper May 2010.
[5] Banday, M.T. Qadri, J.A.Shah, “Study of Botnet and Their Threats to internet Security”, Sprouts:Working papers on Information System, 2009. Available at:http://sprouts.aisnet.org/9-24.
[6] Niels Provos, Thorsten Holz, “Virtual Honeypots: From Botnet Tracking to Intrusion Detection”, See Chapter11- Tracking Botnets. Publisher: Addison Wesley Professional, Publishing 16 June, 2007.
[7] Hossein Rouhani Zeidanloo, Farhoud Hossswinpour and Farhood Farid Etemad “New Approach For Detection of IRC and P2P Botnet” International Journals of Computer and Electrical Engineering, Vol.2, No.6, PP.1029-1038, December, 2010.
[8] Jing Liu, Yang Xiao, Kaveh Ghaboosi, Julia Deng, Jingyuan Zhang, “Botnet: Classification, Attacks, Detection, Tracing, and Preventive Mesaures”, EURASIP Journal on Wireless Communications and Networking, 2008.
[9] Mukesh Kumar, Pothula Sujatha, P. Manikandan, Madarapu, Naresh Kumar, Chetana Sidige and Sunil Kumar Verma “Self-Destructible Concentrated P2P Botnet” IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No.1, PP.401-405, 2011.
[10] Trend Micro: Taxonomy of Botnet Threats ,Available at http://emea.trendmicro.com/imperia/md/content/us/pdf/threats/securiylibrary/botnettaxonomyehitepapernovember2006.pdf.
[11] Mitsuaki Akiyama, Takanori Kawamoto, Masayoshi Shimamura, “A Proposal of metrics for Botnet Detection Based on its Cooperative Behavior”, Ninth Annual International Conference on Privacy, Security and Trust, pp.67-70, 2011.
[12] Anders Orsten Flaglien “Computational Forensics for Botnet Analyzation and Botnet Attack Mitigation”. Available at: http://www.andersof.net/wp-content/uploads/cf_paper-botnets-Andersoflaglien.pdf.
[13] David Dagon, Guofei Gu, Christopher P. Lee “ A Taxonomy of Botnet structures”.
[14] Anna Sperotto, Gregor Schaffrath, Ramin Sadre, “IP Flow Based Intrusion Detection”, in Proc. IEEE Communications Surveys & Tutorials. VOL. 12, no.3 Third Quarter, 2010.
[15] Yukiko Sawaya, Ayumu Kubota, Yutaka Miyake “Detection of Attackers in Services Using Anomalous Host Behavior Based on Traffic Flow Statistics” PSJ International Symposium On Applications and the Internet, PP. 353-359,2011.
[16] Alireza Shahrestani, Maryam Feily Rodina Ahmad, Sureswaran Ramadass “Discovery of Invariant Bot Behavior Through Visual Network Monitoring System” Fourth International Conference on Emerging Security Information, Systems and Technologies,PP.182-188, 2010.
[17] Hsiao-Chung Lin, Chia-Mei Chen, Jui-Yu Tzeng “Flow Based Botnet Detection” IEEE Fourth International Conference on Innovative Computing, Information and control pp.1538-1541, 2009.
[18] Chunyong Yin, Ali A. Ghorbani “ P2P BotNet Detection based on Association Between Common Network Behaviors and Host Behaviors” PP. 5010-5012, 2011.
[19] Yousof Al Hammadi, Uwe Aickelin “Behavioral Correlation for Detecting P2P Bots” Second International Conference on Future Networks, DOI 10. 1109/ICFN 2010.72, PP. 323-327, 2010.
[20] G.Munz and G. Carle “Deep Analysis of Intending Peer-To-Peer Botnet.” 10th IEEE International Symposium on Integrated Network Management , pp.1342-1347 ,2009.
[21] Anna Sperotto, Gregor Schaffrath, Ramin Sadre, Cristian Morariu “An Overview of IP Flow-Based Intrusion Detection”, IEEE Communication Surveys & Tutorials, Vol 12, No. 3, Third Quarter , pp.343-356, 2010.
[22] Mohammed Abdul Qadeer, Mohammad Zahid “Network Traffic Analysis and Intrusion Detection Using Packet Sniffer” Second International Conference on Communication Software and Networks PP.313-317, 2010.
[23] Khaled m. hammouda and Mohamed s. kamel “Hierarchically Distributed Peer-To-Peer Document Clustering Cluster Summarization”, IEEE Transactions on Knowledge and Data Engineering, Vol.21, No.5,PP.681-698, May 2009.
[24] Pinghui Wang, Xiaohong Guan, Tao Qin “P2P Traffic Identification Based on The Signature of Key Packets”, IEEE conference Local Computer Networks, 2009.
[25] Bist, Ankur Singh. "Detection of metamorphic viruses: A survey." Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on. IEEE, 2014.
[26] Bist, Ankur Singh. "Classification and identification of Malicious codes." IJCSE (2012).

[27] Bist, Ankur Singh, and Sunita Jalal. "Identification of metamorphic viruses." Advance Computing Conference (IACC), 2014 IEEE International. IEEE, 2014.
[28] Bist, Ankur Singh. "Hybrid model for Computer Viruses: an Approach towards Ideal Behavior." (2012)
[29] Bist, Ankur Singh. "Pattern matching algorithms for computer virus detection." (2013).
[30] Bist, Ankur Singh. "Fuzzy logic for computer virus detection." IJESRT, ISSN(2014): 2277-9655.
[31] Bist, Ankur Singh. "INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY VIRUS GENERATION KITS: A SURVEY.".

CITE AN ARTICLE

Dwivedi, Sanjeev Kumar, Ankur Singh Bist, and P. K. Chaturvedi. "RECENT TRENDS IN BOTNET RESEARCH." ***INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY*** **6.7 (2017): 280-95. Web. 15 July 2017.**