Botnet Detection by Network Behavior Analysis

Mr. Yogesh Sharma
Research Mentor & Author
Department of Computer Science & Engineering
MAIT, Delhi, India

Nipun Agrawal
Co-Author
Department of Computer Science & Engineering
MAIT, Delhi, India.

Abstract

One of the most possible vulnerabilities to data available over network can be a botnet attack which can cause significant amount of data loss. A botnet attack is a type of malicious attack that utilizes a series of connected computers to attack or take down a network, network device, website or an IT environment. The attack can slow down the network/server, making it busy enough that other legitimate users are unable to access it or temporarily freeze the server. Distributed denial of service (DDOS) is common example of a botnet attack that utilizes a number of botnet devices to send a large number of simultaneous requests/packets to the targeted system. Thus in this paper we collected data sets (i.e. packets travelling in a network) from various sources and merged it to obtain a larger set comprising of benign and malicious traffic. The packets are then analysed to obtain TCP/UDP based flows. Features are then computed for all the flows identified and listed in a feature vector table. We further tried to parallelize the feature computation work using Hadoop map reduce framework. The feature vector table can be further used to train the classifier for segregating the malicious traffic from the benign traffic.

Keywords- Bot, Bot-master, Botnet, P2P, Flows, Feature Vector

I. INTRODUCTION

A bot is an autonomously operating software agent which may be controlled by a remote operator (the bot-master) to perform malicious tasks typically installed onto a victim's computer without the owner's consent or knowledge. [2]. A botnet is a collection of computers connected to the Internet which have been compromised and are being controlled remotely by an intruder via malicious software called bots. [1]
There are three kind of botnet:

A. IRC Based Botnet

Internet Relay Chat (lRC) has been for a while the most prevalent communication scheme among traditional botnets. After infection, the bot will locate and connect with an IRC server. The bot master (the criminal controlling the botnet) will use established IRC command and control (C&C) channels to communicate and control the bots. The bot master will try to keep the bots under control as long as possible. From time to time the bots will connect to the bot master to get new instructions and update their behavior.

B. Http Based Botnet

It's functioning is similar to IRC based botnet except that it uses Http server instead of IRC server. However, both IRC and HTTP-based botnets are vulnerable because they are based on highly centralized architectures; one can disrupt the entire botnet by simply shutting down the IRC or HTTP server.

C. Peer- To- Peer (P2p) Based Botnet

Currently the new trend in botnet communication is toward Peer-to-Peer architectures. Unlike IRC and HTTP-based botnets, in P2P-based botnets, there is no central point to shut down the botnet. Furthermore the bot master can inject commands into any part of the P2P botnet. Because of its elusive nature, it has so far been very difficult to estimate the size of a typical P2P based botnet. To make things worse, encrypted botnets such as Nugache have been able to evade most of the available botnet detection techniques.

Botnet lifecycle unfolds itself in three phases: formation phase, C&C phase, attack phase and post-attack phase. In the formation phase, bot master spreads bots by infecting other machines on the Internet so that they become members of the botnet. In C&C phase, the bots (i.e. infected machines) which are enslaved will receive on regular basis instructions from the bot master. In the attack phase, bots will carry malicious activities based on the received instructions. In the post-attack phase, bot master will try (from time to time) to probe the botnet to get information about active bots and plan for new formation.

All rights reserved by *www.grdjournals.com*

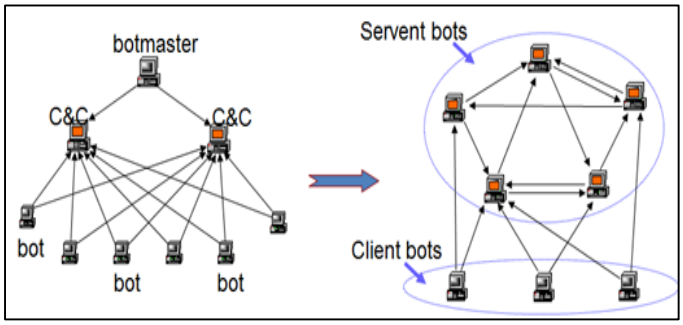

Fig. 1: From centralized botnet to peer-to-peer botnet

II. MOTIVATION

According to report by Damballa, a cyber-security company, few millions of computers in the United States were infected by botnets in 2009.
Following type of malicious activity are performed by a bot installed host

Fig. 2: Malicious attacks

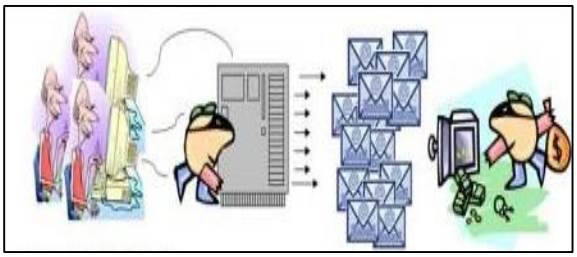

Fig. 3: Sending Virus

Table 1: Earning money from spammers by giving botnet at rent

	Stealing	*Denial of Service Attack*	*Click Fraud*
They send - spam - viruses - spyware	*They steal personal and private information and communicate it back to the malicious user: - credit card numbers - bank credentials - other sensitive personal information*	*Launching denial of service (DoS) attacks against a specified target. Cybercriminals extort money from Web site owners, in exchange for regaining control of the compromised sites.* *More commonly, however, the systems of everyday users are the targets of these attacks.*	*Fraudsters use bots to boost Web advertising billings by automatically clicking on Internet ads*

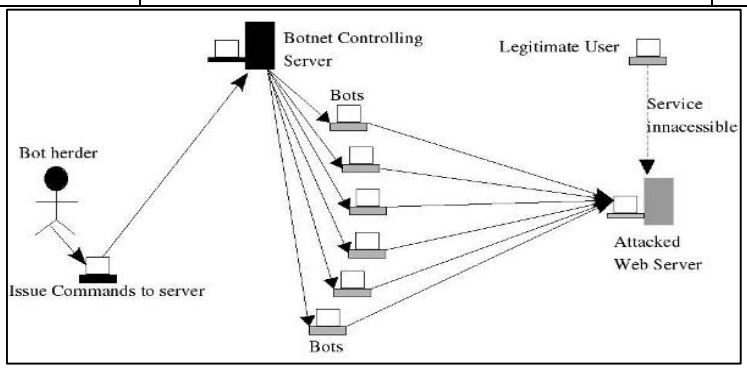

Fig. 4: Denial of Service attack

All rights reserved by *www.grdjournals.com*

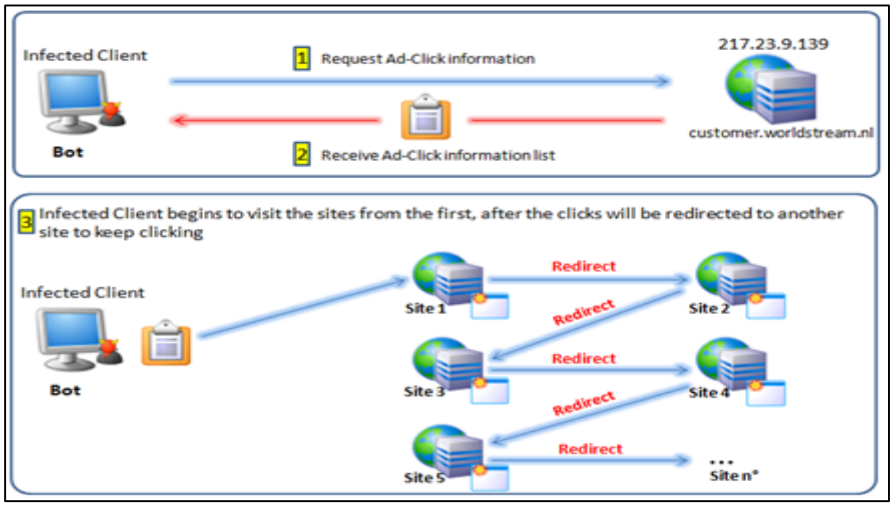

Fig. 5: Click Fraud attack

Therefore, safety and security is the prime concern for network users which requires design and development of efficient techniques for botnet detection.

III. RELATED WORK

A large collection of literature exists for the detection of botnets though interest towards the detection of peer to peer botnets. Furthermore, botnet detection approaches using flow analysis techniques have only emerged in the last few years and of these most examine flows in their entirety instead of smaller time intervals. Most of the literature in this field focuses on analyzing P2P botnet, and characterizing their structure, organization and operation.

Gu et al. presents a general botnet detection framework referred to as BotMiner that is independent of the botnet C&C protocol, structure, and infection model. Likewise the proposed framework targets both centralized IRC and P2P botnets. The working assumption of BotMiner is that bots are coordinated malware that exhibit similar communication patterns and similar malicious behaviours. Hosts with similar communication patterns and those performing similar malicious activities are clustered from captured network flows in the so-called A-plane (activity traffic) and C-plane (C&C communication traffic), respectively. And then, by performing cross-plane correlation across A-plane and C-place clusters,

A key limitation of BotMiner is that by design it targets essentially groups of compromised machines within a monitored network. However, it is common that there is only a single compromised host belonging to the monitored network, although such host may belong to a larger botnet. Under such scenario, BotMiner may not be effective at detecting the compromised host. Another important limitation of BotMiner is the systematic classification of any host that behaves maliciously and exchanges C&C messages as a bot. Although, in general, this may be the case for IRC botnets, it is not always the case for P2P botnets. A peer may behave maliciously and still exchange normal C&C messages as simply being part of a P2P network. In this case, it may only be qualified as a bot if it exchanges covert botnet messages.

Yu et al. proposed a data mining based approach for botnet detection based on the incremental discrete Fourier transform, achieving detection rates of 99% with a false positive rate of 14%. In their work, the authors capture network flows and convert these flows into a feature stream consisting of attributes such as duration of flow, packets exchanged etc. The authors then group these feature streams using a clustering approach and use the discrete Fourier transform to improve performance by reducing the size of the problem via computing the Euclidean distance of the first few coefficients of the transform. By observing that individual bots within the same botnet tend to exhibit similar flow patterns, pairs of flows with high similarities and corresponding hosts may then be flagged as suspicious, and a traditional rule based detection technique may be used to test the validity of the suspicion.
Fedynyshyn et al. [4] have proposed an approach for detecting bots during C&C phase. 17 flow based and host based features are identified that are used as part of a feature vector. Various machine learning techniques, namely Nearest Neighbours Classifier, Linear Support Vector Machine, Artificial Neural Network, Gaussian Based Classifier and Naive Bayes classifier are used to classify feature vectors into three classes i.e. BotNet C&C, non-P2P traffic and normal P2P traffic.

Zhao et al. [5] have used machine learning classification techniques to detect the BotNet activity. A set of 13 attributes is used to capture the characteristics of a single flow for a given time interval. Bayesian network classifier and Decision tree classifiers are used to classify the network traffic into Botnet traffic and non-malicious traffic based on attribute sets.

Our own approach is similar to the one used in last two references listed above. Though in our approach we have introduced new flow attributes which have contributed in improving the accuracy by noticeable amount. Also we have tried to parallelize our code, using hadoop map reduce framework for flow identification and computation of features based on it.

All rights reserved by *www.grdjournals.com*

IV. SCOPE OF PAPER

This paper aims to identify the types of bot that is present in different hosts within a network.
Following types of bot are considered in this paper:-

A. Storm

It is of two type viz. SMTP spam (storm) and UDP (storm). Storm worm is spread via email spam which entices users to click the attachments or to visit some URLs to download the binaries for the purpose of installing the bot on the victim. It seeds botnets based on P2P Overnet (a.k.a Kademila) protocol as C&C. Once a host system is infected by a Storm instance, it will connect to the Over net botnet and become a bot thereafter. Storm sets up by adding a system driver into the host. This driver is injected into 'services.exe', a Windows process. To become part of the botnet, it bootstraps by connecting hundreds of IPs contained in a peer list file hard-coded in the binary. After joining the network, the bot sends out search requests to find a specific secondary injection such as spam template, email harvester, rootkit component, etc. If the target file is successfully located, the bot will download and execute it. Bots can be programmed to obtain different injections in order to perform different tasks. It is also possible for a bot to update itself periodically. The P2P architecture allows each bot to actively seek its task instead of waiting passively for the C&C from a central server.

B. SMTP Spam (Waledac)

Waledac is a worm that is capable of harvesting and forwarding password information. It is capable of receiving commands from a remote server. Commands include instructions on functions to perform (for example, update malware components or send information from the infected computer). The Waledac botnet can be regarded as the successor of Storm Worm. However, Waledac uses a more decentralized store-and-forward communication paradigm and new communication protocols so we had to develop novel techniques to track this botnet.

C. Zeus

It is of two type viz. Zeus and Zeus(C & C). Zeus, ZeuS, or Zbot is Trojan horse computer worm that runs on versions of the Microsoft Windows. While it is capable of being used to carry out many malicious and criminal tasks, it is often used to steal banking information by man-in-the-browser keystroke logging and form grabbing. It is also used to install the Crypto Locker ransomware. Zeus is spread mainly through drive-by downloads and phishing schemes. Zeus is very difficult to detect even with up-to-date antivirus and other security software as it hides itself using stealth techniques. It is considered that this is the primary reason why the Zeus malware has become the largest botnet on the Internet: some 3.6 million PCs are said to be infected in the U.S. alone. Security experts are advising that businesses continue to offer training to users to teach them to not to click on hostile or suspicious links in emails or Web sites, and to keep antivirus protection up to date. Antivirus software does not claim to reliably prevent infection; for example Browser Protection says that it can prevent "some infection attempts"
Features that have been computed for each flow are:

1) Source Port
2) Destination Port
3) Protocol
4) Average Packet Length
5) First Packet Length
6) Total Number of Packets per flow
7) Total Number of Bytes per flow
8) Total Duration of flow
9) Average Inter arrival time of packets in a flow.
10) Number of distinct packet length subsets over number of packets.
11) Total number of incoming packets in the flow.
12) Total number of outgoing packets in the flow

V. PROBLEMS FACED AND SOLUTION ADOPTED

A. Problem 1

Dataset was of 10 GB and program was not able to handle such huge data.
Solution: We processed .pcap file in chunks of 300 milli seconds and considered data packets of a specific subnet. This subnet was chosen as it was a good mix of malicious (all 6 kinds of bot) and non-malicious traffic.

B. Problem 2

Program took 1.5 hour to give the correct result.
Solution: Hadoop was used as a platform to parallelize the flow identification and feature computation for each identified flow which resulted in reduction of computation time.

All rights reserved by *www.grdjournals.com*

VI. ASSUMPTIONS

1) In this paper we have assumed that the dataset abstracted contains only the following kinds of botnet
   a) SMTP spam(storm)
   b) SMTP spam (Waledac)
   c) UDP storm
   d) Zeus
   e) Zeus C&C
2) The information about the machines that generate malicious/non-malicious traffic and their corresponding labels is known beforehand.
3) Text file is taken as input to Hadoop framework. So .pcap file is first converted to .txt format and then provided as input to Hadoop framework.

The paper unfolds in the following steps:-

A. Dataset Understanding

To analyse the network traffic behavior for labeling it into malicious or non – malicious, network trace composed of both types of traffic was desired. Thus to satisfy this requirement we used ISOT Dataset.

1) Overview of ISOT Dataset

The ISOT dataset is the combination of several existing publicly available malicious (comprising of Storm, Waledac & Zeus Botnet) and non-malicious datasets. In this dataset, separate datasets containing malicious traffic from the French chapter of the honeynet project involving the Storm and Waledac botnets respectively were used. To represent non-malicious, everyday usage traffic, we incorporated two different datasets, one from the Traffic Lab at Ericsson Research in Hungary and the other from the Lawrence Berkeley National Lab (LBNL). The Ericsson Lab dataset contains a large number of general traffic from a variety of applications, including HTTP web browsing behavior, World of War craft gaming packets, and packets from popular bit torrent clients such as Azureus. The dataset from the LBNL trace data was also incorporated to provide additional non-malicious background traffic. The LBNL is a research institute with a medium-sized enterprise network.

In order to produce an experimental dataset with both malicious and non-malicious traffic, the above datasets were merged into a single individual trace file via a specific process. First the IP addresses of the infected machines were mapped to two of the machines providing the background traffic. Second, all of the trace files were replayed using the Tcp Replay tool on the same network interface card in order to homogenize the network behavior exhibited by all three datasets.

The final evaluation data produced by this process was further merged with all datasets from the LBNL trace data to provide one extra subnet to even simulate a real enterprise size network with thousands of hosts. The resulted evaluation dataset contains 22 subnets from the LBNL with non-malicious traffic and one subnet (172.16.0.0/16) as illustrated in below figure.

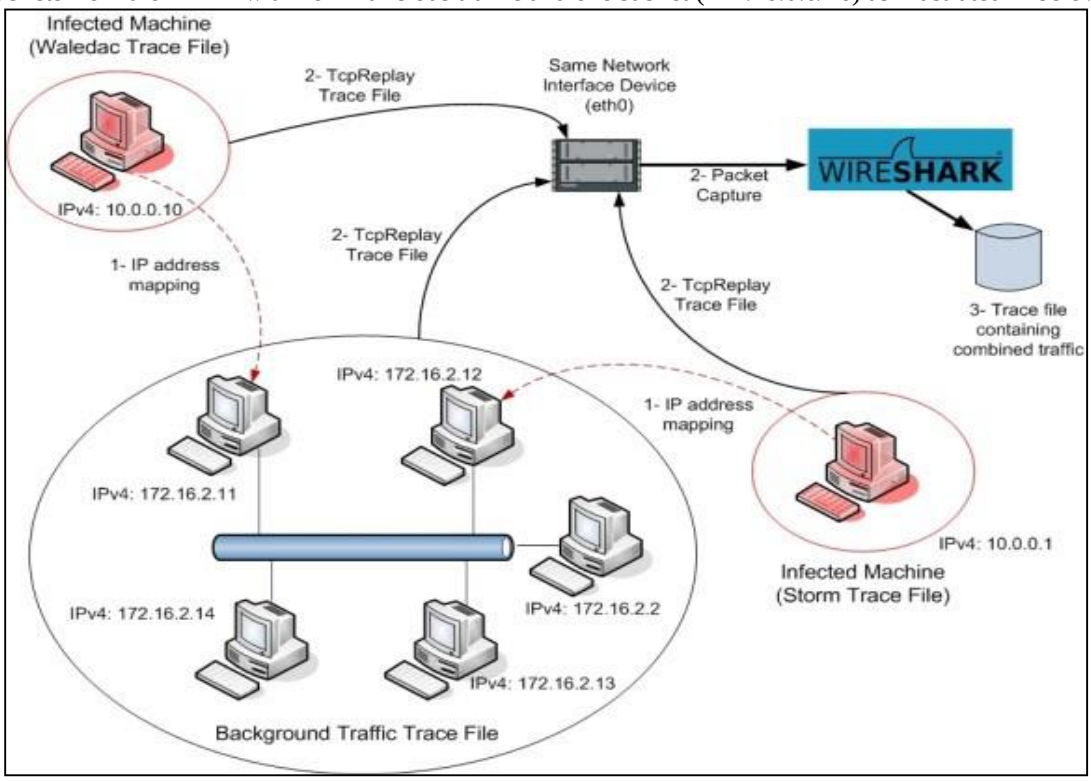

Fig. 6: Dataset merging

All rights reserved by *www.grdjournals.com*

In our paper, we have used the dataset belonging to the specific subset (172.16.0.0/16) for it consists of both malicious and non-malicious traffic.

B. Flow Identification & Flow-Based Feature Computation

Flows were identified based on the following features of the packets, captured in the network trace:

1) Source Ip address
2) Destination Ip address
3) Source Port number
4) Destination Port number
5) Protocol

In our paper, we referred two research papers, listed in the reference section, for identifying the features pertaining to the behaviour of botnet in a network.

C. Network Behavior Classification

By labelling of flows, after identification of network behaviour characteristics, we also need to analyse and classify the network behavioural characteristic in order to isolate the botnet traffic from the normal one. In order to do so, we referred to table3.4 provided originally by ISOT Dataset Overview.

Since we are considering only the subnet 172.16..0/16, thus only the machines in it are considered for classification purpose. This table is used to label each identified flow into type malicious or non-malicious.

Table 2: List of machines that generate malicious/non-malicious traffic and corresponding labels

IP address	*Characteristic of Traffic*	*Flow Label*
172.16.2.11	*Src/Dst MAC BB:BB:BB:BB:BB:BB*	*Malicious/ UDP (Storm)*
172.16.0.2	*Src/Dst MAC AA:AA:AA:AA:AA:AA*	*Malicious/ SMTP Spam (Waledac)*
172.16.0.11	*Src/Dst MAC AA:AA:AA:AA:AA:AA*	*Malicious/ SMTP Spam (Storm)*
172.16.2.2	*Src/Dst MAC AA:AA:AA:AA:AA:AA*	*Non-Malicious*
172.16.2.3	*Normal Src/Dst MAC*	*Non-Malicious*
172.16.2.11	*Normal Src/Dst MAC*	*Non-Malicious*
172.16.2.11	*Normal Src/Dst MAC*	*Non-Malicious*
172.16.2.12	*Normal Src/Dst MAC*	*Non-Malicious*
172.16.2.12	*Src/Dst MAC CC:CC:CC:CC:CC:CC*	*Malicious/ Zeus*
172.16.2.12	*Src/Dst MAC CC:CC:CC:CC:DD:DD*	*Malicious/ Zeus (C & C)*
172.16.2.13	*Normal Src/Dst MAC*	*Non-Malicious*
172.16.2.14	*Normal Src/Dst MAC*	*Non-Malicious*
172.16.2.111	*Normal Src/Dst MAC*	*Non-Malicious*
172.16.2.112	*Normal Src/Dst MAC*	*Non-Malicious*
172.16.2.113	*Normal Src/Dst MAC*	*Non-Malicious*
172.16.2.114	*Normal Src/Dst MAC*	*Non-Malicious*

VII. RESULTS & OBSERVATIONS

The datasets that we considered consists of around 77 crore packets. Our program is taking into consideration packets of a particular subnet only, contributing to about 30, 44,773 packets.

We have classified these 30, 44,773 packets into different flows and computed the feature vector for each identified flow. The output of our code is a text file whose each record correspond to a particular flow with its feature vector.

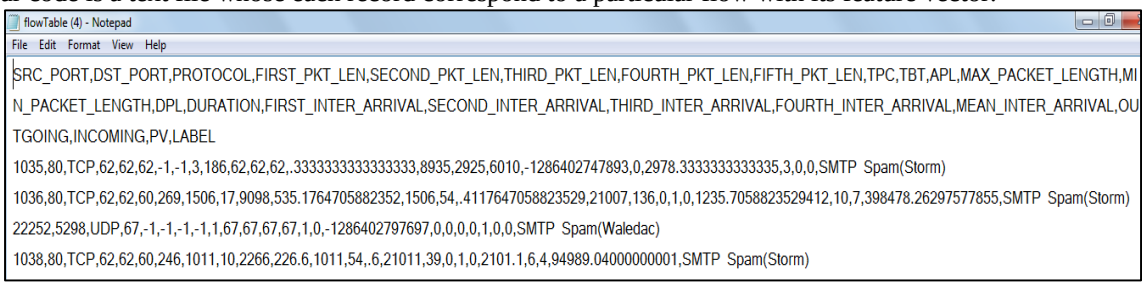

Fig. 7: Output of feature computation code

We installed hadoop on a single node and multi node. We further used hadoop to parallelize the computation and so far we have performed flow identification and feature computation using hadoop. The serial code took around 9264781 milliseconds to identify flows and compute feature vector for each identified flow whereas the code executed on Hadoop framework took around milliseconds for the same.

All rights reserved by *www.grdjournals.com*

VIII. ASSUMPTIONS

We aim to use different Machine Learning techniques for labeling the identified flows. We would apply these techniques on the flow based features derived from [1] and [2] and on our own proposed features and calculate the accuracy for each. We further intend to compare the obtained accuracies through graph. So far file input fomat for Hadoop is a .txt file. We aim to generalize our code by providing .pcap file as an input to Hadoop. We aim to produce a real time network traffic comprising of malicious and non-malicious traffic within the LAN. We will use Wireshark as a tool to capture this traffic.

REFERENCES

[1] S. Saad and W. Lu, "Detecting P2P botnets through network behavior analysis and machine learning," in Proceedings of 9th Annual Conference on Privacy, Security and Trust (PST), IEEE, 2011.
[2] D. Zhao and W. Lu, "Peer to Peer Botnet Detection Based on Flow Intervals," in Information Security and Privacy Research. Springer Berlin Heidelberg, 2012, pp. 87-102
[3] P.Narang, J.M. Reddy and C. Hota, "Feature selection for detection of peer-to-peer botnet traffic", In Proceedings of the 6th ACM India Computing Convention ACM, 2013
[4] Gregory Fedynyshyn, Mooi Choo Chuah, and Gang Tan," Detection and Classification of Different Botnet C&C Channels" www.cse.psu.edu/~gxt29/paper
[5] Sherif Saad, Issa Traore, Ali Ghorbani, Bassam Sayed, David Zhao, Wei Lu, John Felix, Payman Hakimian," Detecting P2P botnets through network behavior analysis and machine learning"

Website References

[6] http://resources.infosecinstitute.com/botnets-and-cybercrime-introduction/
[7] http://in.norton.com/botnet
[8] http://ilookbothways.com/page/4/
[9] http://mac-internet-security-software-review.toptenreviews.com/how-do-i-know-if-my-computer-is-a-botnet-zombie-.html
[10] http://www.chmag.in/ro/node/420
[11] http://www.uvic.ca/engineering/ece/isot/assets/docs/isot-datase.pdf

All rights reserved by *www.grdjournals.com*