ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 3, March 2013

Mega Behavioral Based Detection and Defending Techniques

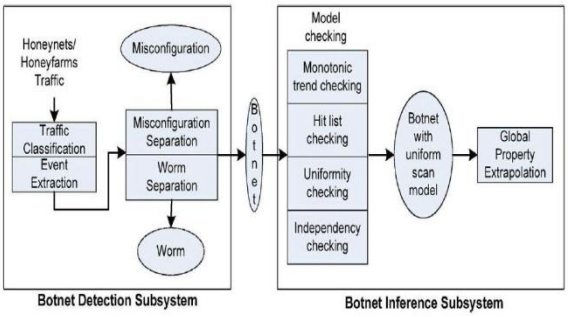
Vijayashankar.J[1], B.vinodhini[2], S.karthik[3]
PG scholar[1], Asst.Proffesor[2], Dean& Proffesor[3]
Department of Computer Science And Engineering,
SNS College Of Technology, Coimbatore.
Hand Phone: 9994431402

Abstract- **Botnet is the term in which a malware instance runs autonomously and automatically on a compromised computer (zombie) without owner's consent. IDS provides timely information on security risks to protect the network systems from malware campaigns and mitigate against new software vulnerabilities. Initially massive denial-of-service (DoS) attacks had been launched against multiple banks, but declined to comment further. DoS attacks seek to disrupt websites and other computer systems at the targeted organization by overwhelming their networks with computer traffic. ISP provider's main responsibility is to provide the better QoS and protect network from intruders. The criminals and botnet sellers are the power of distributed computing to anonymously url spam, recruit other zombies, information harvesting or commit DoS. The goal is mainly to give a brief information about the latest botnet trends which include botnet architectures used, botnet attacks and latest botnet behaviors.**

Index Terms **– BOTNET, DOS ,IDS,SPAM**

I.INTRODUCTION

Botnets are often named after their malicious software name, there are typically multiple botnets in operation using the same malicious software families, but operated by different criminal entities[1]. While the term "Botnet" can be used to refer to any group of
bots, such as IRC bots, this word is generally used to refer to a collection of
compromised computers (called zombie computers) running software, usually installed via drive-by downloads exploiting web browser vulnerabilities, worms, Trojan horses, or backdoors, under a common command-and-control infrastructure A botnet's originator ("bot herder" or "bot master") can control the group remotely, usually through a means such as IRC, and usually for nefarious purposes. Individual programs manifest as IRC "bots". Often the command-and-control takes place via an IRC server or a specific channel on a public IRC network. This server is known as the command-and-control server ("C&C").More experienced botnet operators program their own commanding protocols from scratch. The constituents of these protocols include a server program, client program for operation, and the program that embeds itself on the victim's machine (bot).



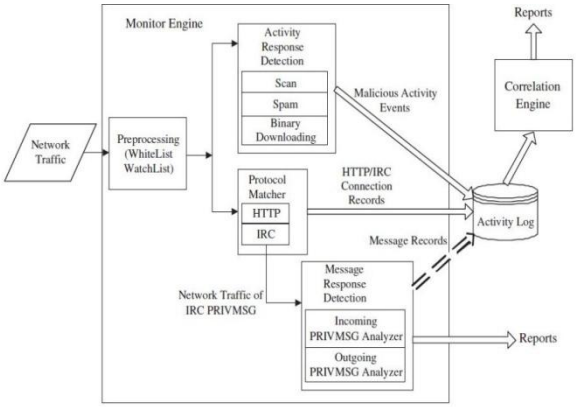
Botnet System Architecture

All three of these usually communicate with each other over a network using a unique encryption scheme for stealth and protection against detection or intrusion into the botnet network. A bot typically runs hidden and uses a covert channel (e.g. the RFC 1459 (IRC) standard, twitter or IM) to communicate with its C&C server. Generally, the perpetrator of the botnet has compromised a series of systems using various tools (exploits, buffer overflows, as well as others; see also RPC). Newer bots can automatically scan their environment and propagate themselves using vulnerabilities and weak passwords. Bot-herder configures initial bot parameters such as infection vectors, payload, C&C details

- Register a DDNS
- Register a static IP
- Bot-herder launches or seeds new bot(s)
- Bots spread
- Causes an increase of DDoS being sent to the victim
- Losing bots to rival botnets

Denial-of-service attacks where multiple systems autonomously access a single Internet system or service in a way that appears legitimate, but much more frequently than normal use and cause the

system to become busy. The term “Bot” is nothing but a derived term from “or-bot” which is a generic term used to describe a script or sets of scripts designed to perform predefined function in automated fashion. Botnet is the collections of bots or collection of compromised computers that are remotely controlled by its Bothered. Botnets shows the trace of existence for several years ago, Botnet have only recently sparked the interest of the research community[2]. Bot activity occurred mostly for sport, under the radar of law enforcement and the enterprise leadership. Now, botnets of more than a million compromised computers are found regularly in the wild, although they usually run in packs of 10 to 20,000 to avoid detection, according to security researchers at Verizon business. Traffic analysis, packet scanning and signature analysis are still reactive (relying on known behavior and signatures), but they’re the only way to detect and stop bots at the network level. Network Intelligence Initiative, Verizon has set up collectors on its backbone devices and core routers to do flow and signature analysis in search of bot activity on its global network. For example, timed pings at regular intervals would be a sign of bot and master staying in contact with each other. Or scans against "dark space" (unassigned IP addresses) indicate a bot scanning engine looking for more machines to infect.



BOTSNIFFER ARCHITECTURE

Ultimately, the goal is to use traffic and packet analysis to follow the herder’s commands back to the bot operator and shut it down. But shutting these bot operators down is not so easy. When we identify these hosts, some of them are overseas, a lot of them in the former Soviet Union and China, so we can’t get the controllers shut down because laws are different in those countries. But when we find bot controllers in the U.S. and we report them, most ISPs will work with us to shut those controllers down.

BOTNET LIFE CYCLE CREATION:First, the botmaster develops his bot software, often reusing existing code and adding custom features. He might use a test network to perform dry runs before deploying the bot in the wild. Infection - There are many possibilities for infecting victim computers, including the following four. Once a victim machine becomes infected with a bot, it is known as a zombie.

- SOFTWARE VULNERABILITIES: The attacker exploits vulnerability in a running service to automatically gain access and install his software without any user interaction. This was the method used by most worms, including the infamous Code Red and Sesser worms.
- DRIVE-BY DOWNLOAD: The attacker hosts his file on a Web server and entices people to visit the site. When the user loads a certain page, the software is automatically installed without user interaction, usually by exploiting browser bugs, misconfigurations, or unsecured ActiveX controls.
- TROJAN HORSE: The attacker bundles his malicious software with seemingly benign and useful software, such as screen savers, antivirus scanners, or games. The user is fully aware of the installation process, but he does not know about the hidden bot functionality.
- EMAIL ATTACHMENT: Although this method has become less popular lately due to rising user awareness, it is still around. The attacker sends an attachment that will automatically install the bot software when the user opens it, usually without any interaction. This was the primary infection vector of the ILOVEYOU email worm from 2000. The recent Storm Worm successfully used enticing email messages with executable attachments to lure its victims.

RALLYING - After infection, the bot starts up for the first time and attempts to contact its C&C server(s) in a process known as rallying. In a centralized botnet, this could be an IRC or HTTP server, for example. In a P2P botnet, the bots perform the bootstrapping protocol required to locate other peers and join the network. Most bots are very fault-tolerant, having multiple lists of backup servers to attempt if the primary ones become unavailable. Some C&C servers are configured to immediately send some initial commands to the bot (without botmaster intervention). In an IRC botnet, this is typically done by including the commands in the C&C channel’s topic. Waiting - Having joined the C&C network, the bot waits for commands from the botmaster. During this time, very little (if any) traffic passes between the victim and the C&C servers. In an IRC botnet, this traffic would mainly consist of periodic keep-alive messages from the server.

EXECUTING - Once the bot receives a command from the botmaster, it executes it and returns any results to the botmaster via the C&C network. The supported commands are only limited by the botmaster's imagination and technical skills. Common commands are in line with the major uses of botnets: scanning for new victims, sending spam, sending DoS floods, setting up traffic redirection, and many more.

DOMAIN GENERATION ALGORITHMS: are a more recent addition to bot agents. They create a dynamic list of multiple FQDN's each day, which are then polled by the bot agent as it tries to locate the C&C infrastructure. Since the created domain names are dynamically generated in volume and typically have a life of only a single day, the rapid turnover makes it very difficult to investigate or block every possible domain name.

BLIND PROXY REDIRECTION: Both IP Flux and Domain Flux provide advanced levels of redundancy and resilience for the C&C infrastructure of a botnet. However, botnet operators often employ a second layer of abstraction to further increase security and failover – blind proxy redirection. Redirection helps disrupt attempts to trace or shutdown IP Flux service networks. As a result, botnet operators often employ bot agents that proxy both IP/domain lookup requests and C&C traffic.

CLICK FRAUD: is the user's computer visiting websites without the user's awareness to create false web traffic for the purpose of personal or commercial gain.

ADWARE: exists to advertise some commercial entity actively and without the user's permission or awareness, for example by replacing banner ads on web pages with those of another content provider.

SPYWARE: is software which sends information to its creators about a user's activities – typically passwords, credit card numbers and other information that can be sold on the black market. Compromised machines that are located within a corporate network can be worth more to the bot herder, as they can often gain access to confidential information held within that company. There have been several targeted attacks on large corporations with the aim of stealing sensitive information, one such example is the Aurora botnet.

E-MAIL SPAM: are e-mail messages disguised as messages from people, but are either

ACCESS NUMBER REPLACEMENTS: are where the botnet operator replaces the access numbers of a group of dial-up bots to that of a victim's phone number. Given enough bots partake in this attack, the victim is consistently bombarded with phone calls attempting to connect to the internet. Having very little to defend against this attack, most are forced into changing their phone numbers (land line, etc).

Communication Protocols A communications protocol is a system of digital message formats and rules for exchanging those messages in or between computing systems and in telecommunications. So studying about communication protocols can help us determine the origins of a Botnet attack and decode conversations between the bots and the botmasters. Communication protocol can be classified in three different categories:

IRC PROTOCOL: A most common protocol used by botmasters to communicate with their Bots. IRC protocol mainly designed for one to many conversations but can also handle one to one, which is very useful for Botmasters control their Botnet. However, security devices can be easily configured to block IRC traffic.

Weaknesses of IRC bots:

- Usually unencrypted
- Easy to get into, take over or shut down
- Due to the dependability more on C&C Server, Single point of failure is there .

HTTP protocol - Generally HTTP protocol is a popular Botnet due to its communication method by sending message as HTTP response and HTTP GET response to perform attack which is difficult to be detected. So Using the HTTP protocol, Botnet usually bypass security devices.

Weaknesses of HTTP based bots:

- Due to the dependability more on C&C Server, single point of failure is there.
- Bypass attack possible

WEAKNESSES OF P2P BASED BOTS:

- Strict Dependent ability on previous or others nodes
- These will not generate a sound Botnet
- Not mature
- If these have poor connectivity then easily traced
- Compared to HTTP Botnet, these have no hardly encryption /authentication code
- For large number of nodes, creates a complex structure and generates a large amount of traffic
- WASTE P2P protocol is not scalable across a large network.

In last several years, Botnets such as Slapper, Sinit, Phatbot, and Nugache have implemented different kinds of P2Pcontrol architectures. They have shown several advanced designs. For example, Sinit uses public key cryptography for update authentication but has poor connectivity.

II.BOTNET DETECTION AND DEFENDING TECHNIQUES

It is important that the analysis of structure and distribution is enabled among the ISPs. It is also crucial to understand botnets' behavior based on information that is gathered. The function of monitoring and controlling the network is available to every ISP but can be exploited[3]. The network monitoring and controlling is done through Security Authority (SA) via collecting and attack information on ISPs. Boot zombies bypass the ISP using a DNS sinkhole that accepts command/control traffic. However, traffic data from these sinkholes and analyzed to be shared internationally. These traffic data collected LAN and backbone is then sent to the detection module.

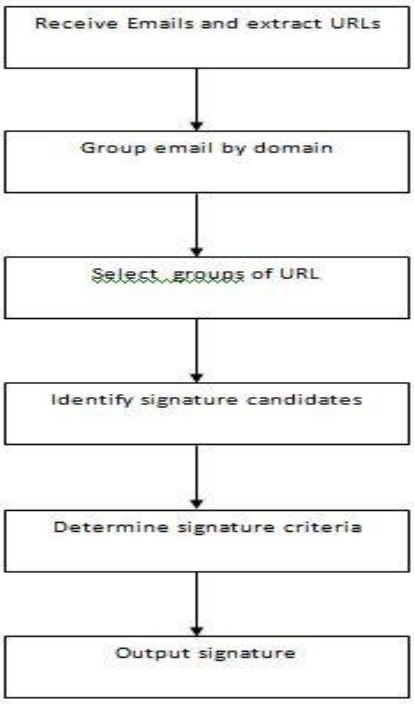
NETWORK-BASED OBSERVATION AND FILTERING UNLIKELY FLOWS:

Most computationally intensive detection of botnet coordination is done on a dramatically reduced traffic set. Filtering flows to reduce the botnet search space. Characterize IRC flows (which are also brief and interactive) to identify how we can separate the C2 channel from other Internet traffic. For data reduction, they convert the sequence of packets into flow summaries, and then identify a suspect infrastructure. Collect and analyze forensic archive of packet- level of data.

EMAIL SHAPE DETECTION Botnet detection mechanism can be based on the neither email " shape" neither analysis that relies on neither content nor reputation analysis. Email can be characterized by mimicking human visual inspection. A set of email shapes are derived and then they used to generate a botnet signature. Detect the presence spamming botnet that sent email.

ANALYZE EMAIL CLIENT INTERACTIONE mail has the ability to provide a stealthy C&C channel. Messages can be cleverly crafted to include a botnet mechanism that can run without user inter- action at their email clients and is invisible to detection mechanism. To evaluate the ability of bot- masters, two C&C channels are designed to employ encoded commands in emails. Subsequent communications over email communication channels between bots and botmasters permits access without identification or blockage.

SYSTEM ARCHITECTURE DESIGN



III.SYSTEM ARCHITECTURE DESIGN DESCRIPTION

Existing Intrusion detection methods usually rely on precise prediction of failure probabilities for a task on a resource in a certain time slot. This prediction is very hard to achieve and requires not only a deep knowledge of the behavior of the target infrastructure, but also years of trace data of the specific system. To find a compromise balance between these two complementary techniques, new algorithm called Resubmission Impact (RI) that tries to establish a metric describing the impact of resubmitting a task to the overall execution time of a workflow application, and to adjust the replication size of each task accordingly[4]. A generic method for fault tolerance which can be immediately applied to any workflow in any distributed computing environment, even if no historic trace data to build an environment-specific fault model is available.

The idea behind RI is to establish a metric describing the impact of having to resubmit a task on the execution time of a workflow application. The RI heuristic is able to schedule workflows with a high success rate while consuming a reasonably low amount of resources. However, since RI schedules the full workflow before it is executed, it cannot react

to unexpected periods of high failures during execution and is therefore unable to adjust its replication size accordingly. This can lead to situations where the replication can fail and many tasks have to rely on resubmission. The workflow enactment responsible for submitting the tasks and transferring data according to the control flow and data flow dependencies.

The new rescheduling heuristic is invoked if the enactment engine discovers that the real workflow make span is too far behind the scheduled workflow execution plan, which implies a heightened probability of missing the soft deadline presented to the user. The role of the rescheduling heuristic is to decrease the probability of violating soft deadlines by adjusting the amount of replication. The existing system does not rely on resource failure prediction that is hard to achieve even with years of historic failure trace data of the target environment. It consists of

- It makes intensive use of additional resources during replication.
- It increases the overall workflow make span if there are not enough free resources available.

IV.PROPOSED SYSTEM

DEVELOP AN INTRA NETWORK FOR DOMAIN MESSAGING SERVICE

In this module we have to develop an intra network environment to mail service. We deploy the mail server over the network. Using java client and server environment create this intra network environment. It can implements in limited no of system access in the environment.

CREATE THE DATABASE OF HISTORY AND BEHAVIOR OF BOTNETS

In this module create the database with the history and behavioral list based on records in the world till now which are all they detected. It can be helpful to monitoring the network and also helpful for identifying the botnet. Its nature like that antivirus tool so far. Using the MYSQL database we have to develop one database for track records and based on that record it can be done. Implementation of monitoring algorithm and defending algorithm IRC PRIVMSG messages for further correlation analysis. Here mainly uses two anomaly detection modules, namely, the abnormally high scan rate and weighted failed connection rate. Here uses a new detector for spam behavior detection, focusing on detecting MX DNS query (looking for mail servers) and SMTP connections (because normal clients are unlikely to act as SMTP servers).We note that more malicious activity response behaviors can be defined and utilized. This project first groups the clients according to their destination IP and port pair. That is, clients that connect to the same server will be put into the same group. This project performs a *group analysis* of spatial-temporal correlation and similarity.

ALGORITHM 1. RESPONSE-CROWD-DENSITY-CHECK ALGORITHM

We use a binary random variable Yi to denote whether the ith response crowd is dense or not. Let us denoteH1 as the hypothesis “botnet”,H0 as “not botnet”. We define Pr(Yi|H1) = θ1 and Pr(Yi|H0) = θ0, i.e., the probability of the ith observed response crowd is dense when the hypothesis “botnet” is true and false, respectively. Clearly, for a botnet, the probability of a dense crowd (θ1) is high because bots are more synchronized than humans. On the other hand, for a normal (non-botnet) case, this probability (θ0) is really low. If we observe multiple response crowds, we can have a high confidence that the group is very likely part of a botnet or not part of a botnet. we utilize a SPRT (Sequential Probability Ration Testing) algorithm , to calculate a comprehensive anomaly score when observing a sequence of crowds. TRW is a powerful tool in statistics and has been used in port scan detection and spam laundering detection[5,6]. By using this technique, one can reach a decision within a small number of rounds, and with a bounded false positive rate and false negative rate. we want to calculate the likelihood ratio _n given a sequence of crowds observed Y1, ..., Yn. Assume the crowds Yis’ are i.i.d. independent and identically-distributed), we have

n = ln Pr(Y1, ..., Yn|H1) Pr(Y1, ..., Yn|H0) = ln Qi Pr(Yi|H1) Qi Pr(Yi|H0) =X I ln Pr(Yi|H1) Pr(Yi|H0)

According to the TRW algorithm, to calculate this likelihood _n, we are essentially performing a threshold random walk. The walk starts from the origin (0), goes up with step

Length ln_1 _0

When Yi = 1, and goes down with

step length ln 1−_1

1−_0

when Yi = 0. Let us denote α and β

the user-chosen false positive rate and false negative rate, respectively. If the random walk goes up and reaches the threshold B = ln 1−__ , this is likely a botnet, and we accept the hypothesis “botnet”, output an alert, and stop. If it goes down and hits the

threshold A = ln _1−_, it is likely not a botnet. Otherwise, it is pending and we just watch for the next round of crowd.

First, it requires observing multiple rounds of response crowds. If there are only a few response behaviors, the accuracy of the algorithm may suffer. In practice, we find that many common commands will have a long lasting effect on the activities of bots.

Second, sometimes not all bots in the group will respond within the similar time window, especially when there is a relatively loose C&C. One solution is simply to increase the time window for each round of TRW.

ALGORITHM 2. RESPONSE-CROWD-HOMOGENEITY-CHECK ALGORITHM

A homogeneous crowd means that within a crowd, most of the members have very similar responses. For example, the members of a homogeneous crowd have message response with similar structure and content, or they have scan activities with similar IP address distribution and port range[8,9]. We note that we currently implement this algorithm only for message response analysis. In this enhanced algorithm, Yi denotes whether the ith crowd is homogeneous or not. We use a clustering technique to obtain the largest cluster of similar messages in the crowd, and calculate the ratio of the size of the cluster over the size of the crowd. If this ratio is greater than a certain threshold, we say Yi = 1; otherwise Yi = 0. There are several ways to measure the similarity between two messages (strings) for clustering.We require that the similarity metric take into account the structure and context of messages. Thus, we choose DICE coefficient (or DICE distance) as our similarity function. DICE coefficient is based on n-gram analysis, which uses a sliding window of length n to extract substrings from the entire string. For a string X with length l, the number of n-grams is |ngrams(X)| = l − n + 1. Dice coefficient is defined as the ratio of the number of n-grams that are shared by two strings over the total number of n-grams in both strings:

Dice(X, Y) = 2|ngrams(X) \ ngrams(Y)||ngrams(X)| + |ngrams(Y)|We choose n = 2 in our system, i.e., we use Bigram analysis. We also use a simple variant of hierarchical clustering technique. In order to make a decision that a crowd is part of a botnet, the expected number of crowd message response.

Rounds we need to observe is: where α and β are user-chosen false positive and false negative probabilities, respectively. Similarly, if the crowd is not part of a botnet, the expected number of crowd message response rounds to make a decision.

PROPSED SYSTEM CORE CONCEPT

In this module Bayesian classifiers work by correlating the use of tokens (typically words, or sometimes other things), with spam and non-spam e-mails and then using Bayesian inference to calculate a probability that an email is or is not spam. Bayesian spam filtering is a very powerful technique for dealing with spam, that can tailor itself to the email needs of individual users, and gives low false positive spam detection rates that are generally acceptable to users.

Bayesian email filters take advantage of Bayes' theorem. Bayes' theorem is used several times in the context of spam:

- a first time, to compute the probability that the message is spam, knowing that a given word appears in this message;
- a second time, to compute the probability that the message is spam, taking into consideration all of its words (or a relevant subset of them);
- sometimes a third time, to deal with rare words.
- Computing the probability that a message containing a given word is spam

Let's suppose the suspected message contains the word "replica". Most people who are used to receiving e-mail know that this message is likely to be spam, more precisely a proposal to sell counterfeit copies of well-known brands of watches. The spam detection software, however, does not "know" such facts, all it can do is compute probabilities.

The formula used by the software to determine that is derived from Bayes' theorem

$$Pr(S|W) = \frac{Pr(W|S).Pr(S)}{Pr(W|S).Pr(S) + Pr(W|H).Pr(H)}$$

…… Equation (4.2.1)

where:

Pr(S|W) is the probability that a message is a spam, knowing that the word "replica" is in it;

- Pr(S) is the overall probability that any given message is spam;
- Pr(W|S) is the probability that the word "replica" appears in spam messages;
- Pr(H) is the overall probability that any given message is not spam (is "ham");
- Pr(W|H) is the probability that the word "replica" appears in ham messages.

The filters that use this hypothesis are said to be "not biased", meaning that they have no prejudice regarding the incoming email. This assumption permits simplifying the general formula to:

$$Pr(S) = \frac{Pr\ (W|S)}{Pr(W|S) + Pr\ (W|H)}$$

-- Equation(4.2.2)

This quantity is called "spamicity" (or "spaminess") of the word "replica", and can be computed. The number Pr(W|S) used in this formula is approximated to the frequency of messages containing "replica" in the messages identified as spam during the learning phase. Similarly, Pr(W|H) is approximated to the frequency of messages containing "replica" in the messages identified as ham during the learning phase[7]. For these approximations to make sense, the set of learned messages needs to be big and representative enough. It is also advisable that the learned set of messages conforms to the 50% hypothesis about repartition between spam and ham, i.e. that the datasets of spam and ham are of same size.

GRAPH MONITOR OF THE SYSTEM INFECTION

This module that system malware infection monitor with the graph with the system .This module create the report based on the statistical analysis basis track record. It can be provide the visual malware infection report to client as well as server machines. Also it can be add the bot behavior to the list of botnet records. Periodic software updates providence for lists of botnets adorns and also we have to provide periodic updates for lists of botnets attacks and it's behavior. It can automatically check for updates within five days.

V.CONCLUSION AND FUTURE WORK

Botnet detection is a relatively new and a very challenging research area. In this project, simple botnet detection and defending scheme used for spam and DDOS attacks. This IDS, employs several correlation and similarity analysis algorithms to examine network traffic. In particularly NAVIE byes algorithm used here to identifies the crowd of hosts that exhibit very strong synchronization and correlation in their responses and activities as bots of the same botnet. It has promising detection accuracy with very low false positive.

VI.REFERENCES

[1] A. Karasaridis, B. Rexroad, and D. Hoeflin, "Wide-scale Botnet detection and characterization," in Proc. 1st Workshop on Hot Topics in Understanding Botnets, 2007.
[2] A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers," in Proc. ACM SIGCOMM, 2006.
[3] A. Schonewille and D. van Helmond, "The domain name service as an IDS", Research Project for the Master System-and Network Engineering at the University of Amsterdam, 2006.
[4] Brett Stone-Gross, Marco Cova, Lorenzo Cavallaro, Bob Gilbert, Martin Szydlowski, Richard Kemmerer, Christopher Kruegel, and Giovanni Vigna, "Your Botnet is My Botnet: Analysis of a Botnet Takeover", 2009.
[5] Chao Li, Wei Jiang, Xin Zou,"Botnet: Survey and Case Study", 4th International Conference on Innovative Computing, Information and Control, 2009.
[6] C. Kalt, "Internet Relay Chat: Client Protocol," Request for Comments (RFC) 2812 (Informational), April 2000.
[7] C. Mazzariello, "IRC traffic analysis for Botnet detection", In Information Assurance and Security, 2008. ISIAS‟ 08. Fourth International Conference on, pages 318–323, 2008.
[8] D. Dagon, G. Gu , C.P. Lee, W. Lee, "A Taxonomy of Botnet Structures," in Proc. 23rd Annual Computer Security Applications Conference (ACSAC 2007), 2007, pp. 325-339.
[9] D. Dagon, "Botnet detection and response", In OARC Workshop, 2005.
[10] Evan Cooke, Farnam Jahanian, Danny Mcpherson, "The Zombie Roundup, Understanding, Detecting, and Disrupting Botnets", IEEE, 2005.
[11] E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: Understanding, detecting, and disrupting botnets," in Proc. Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI‟ 05), 2005, pp. 39-44.
[12] G. Gu, J. Zhang, and W. Lee, "Botsniffer: Detecting Botnet command and control channels in network traffic," in Proc. 15th Annual Network and distributed System Security Symposium (NDSS‟ 08), 2008.

VII.AUTHORS PROFILE

VIJAYASHANKAR.J is doing ME(software engineering) in SNS college of technology, coimbatore, he also pursued MCA in kgisl-iim, coimbatore. His under graduation in shree amman arts and science college, chittode. He is also pusured his MBA in bharathiyar university. He is interested in networking and cyber security.

VINODHINI.B is presently an Assistant Professor in the Department of Computer Science & Engineering, SNS College of Technology, affiliated to Anna University- Coimbatore, Tamilnadu, India. She received the M.E degree from the Avinashilingam University, Coimbatore and currently doing Ph.D in Anna University-Coimbatore. Her research interests include Computer Networks, Mobile Computing and Wireless Communications. She published papers in International and National Conferences.

Professor Dr.S.Karthik is presently Professor & Dean in the Department of Computer Science & Engineering, SNS College of Technology, affiliated to Anna University- Coimbatore, Tamilnadu, India. He received the M.E degree from the Anna University Chennai and Ph.D degree from Ann University of Technology, Coimbatore. His research interests include network security, web services and wireless systems. In particular, he is currently working in a research group developing new Internet security architectures and active defense systems against DDoS attacks. Dr.S.Karthik published more than 35 papers in refereed international journals and 25 papers in conferences and has been involved many international conferences as Technical Chair and tutorial presenter. He is an active member of IEEE, ISTE, IAENG, IACSIT and Indian Computer Society.