Botnets:Architecture and Detection Techniques

Ashwin Yadav*
Department of Computer Engineering
Thakur College of Engineering and Technology
Mumbai, India
ashwindyadav198@gmail.com

Ishita Sharma*
Department of Computer Engineering
Thakur College of Engineering and Technology
Mumbai, India
ishsecure2997@gmail.com

Vidyadhari R.Singh
Department of Computer Engineering
(Assistant Professor)
Thakur College of Engineering and Technology
Mumbai,India
Vidya2686@gmail.com

***Abstract*— This paper explores the various aspects of a botnet, the newest and most efficient tool to launch Distributed Denial of Service attacks. They threaten as well as impair the network infrastructure of an organization. With the help of survey, we try to explore many use-cases of botnets and its quirks. We investigate the architectures norms used by malicious actors to establish botnets. Citing the dormant nature of botnets that is they could be present on a victim's computer system without the knowledge of the victim, in lieu of the same, some botnet detection techniques are also elaborated.**

Keywords—DDOS, IOT

I. INTRODUCTION

The zombies are roaming all around the world and you are one of the few who remain uninfected well it's a fairly good situation than being even unknown of the fact that zombie exists, the similar scenario can be seen in the digital world as well, they are even more scarier and perilous than the zombies of the physical world. Zombies of the digital world control the computer systems owned by the humankind to launch cyberattacks on individuals or organizations with an aim to steal confidential data or completely shut-down their systems. The motive of the attacks is to steal or destroy data, as data is the new oil with which today's modern and fast paced world functions. The zombies are also called as 'bots' which is derived from the word robots that again comes from a Czech word *robota* which means *forced labor*. A malicious actor plants a trojan or worm in thousands of systems remotely that can be achieved through attack vectors like phishing attacks or by sending a malicious link. Once this trojan or virus is planted on the target system the system gets under the control of the attacker and executes malicious instructions issued by the attacker to launch destructive DDOS attacks. Such an army of computers controlled by the attacker is popularly termed as the *botnet.* Any device which has an active internet connection can be controlled by an attacker to make it a part of a botnet. This is the reason why IOT devices are being used by attackers to form botnets because of their increasing demand as well as vulnerable or weak security.

A breathtaking research which required a fortune conducted by an anonymous researcher in 2012 while condemning the action of renowned IT companies demonstrated the weakness of the factory-made devices with pre-configured security vulnerability. Further exploiting the weakness of millions of devices he conducted an Internet survey by making sense out of all the data traffic throughout the world and put it online for anyone to see. Yet this sort of vulnerabilities still exists in fact with the rise of Internet of Things it had gotten worse. Though it was not done with any malign intent as the researcher conveyed he proved his point through such action which is nowhere near small-scale.

Mirai, a botnet that got popular after the attack in 2016 which targeted a large number of vulnerable IOT devices to launch a massive DDOS attack on a DNS service company Dyn. This attack launched in 2016 by a student at Rutgers University and his fellow friends is very intriguing due to the fact that the mitigation services for this attack were also offered by this student and his friends. Mirai encapsulated some clever techniques, including the list of hardcoded passwords.[12]

Mirai scans the internet for IOT device that runs on ARC processors. These devices run a stripped down version of Linux operating system. If the default username-and-password combination is not changed, Mirai is able to log into the device and thus infect it. Investigation of the attack uncovered 49,657 unique IPs which hosted Mirai-infected devices. These were mostly CCTV cameras—a popular choice of DDoS botnet herders. Other victimized devices included DVRs and routers.[13]

Overall, IP addresses of Mirai-infected devices were spotted in 164 countries. Mirai's command and control code (C&C) was coded in Go, while its bots are coded in C.

This botnet had a protocol to locate and compromise IOT devices to further grow the botnet and launch an attack like DDoS based on instructions received from a remote C&C.

To realize its recruitment function, Mirai performs wide-ranging scans of IP addresses. The objective of these scans is to find vulnerable IoT devices that could be remotely accessed through easily discoverable login credentials—usually default usernames and passwords such as admin/admin.

Mirai utilizes a brute force technique for guessing passwords, such brute forcing attacks are also known as the dictionary attacks. One of the most intriguing features of the Mirai botnet was a clever technique that avoids scanning certain IP addresses. These range of IP addresses are hardcoded into the Mirai botnet .These IP addresses include the US postal service, Department of Defense, Internet authority of assigned names and numbers (IANA), and IP ranges belonging to Hewlett Packard and General Electric. These list of IP addresses is popularly known as the "*Don't mess with list*".

One more fascinating thing about Mirai is its "territorial" nature. The malware holds several killer scripts meant to extirpate other worms and Trojans, as well as prohibiting

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

remote-connection attempts on the hijacked device. While DDoS attacks from Mirai botnets can be mitigated, there's no way to avoid being targeted.

II. Network Architecture

There are various possible architectures of a botnet that are designed by malicious actors also known as "the attackers" to orchestrate large scale attacks (large in terms of both the number of machines compromised and the overall impact of the attack).The aim of designing these architectures is basically evading detection as well as defeating the motives of the defenders to mitigate the loss caused by these networks or entirely shut down these networks. While we are at work defending our networks from botnets, the "Bot-masters" are defending their "destructive nets". Bot-masters are individuals who can either be the controller or the designer of botnets. It is not necessary that a Bot-master is a designer, believe it or not, botnets are available for sale and rent as well! from you know where. Thus, a Ready Service available for those who want to fulfill their malicious purposes.

Now, diving deep into the two major architectures used by the Bot-masters to fulfill their purposes. The two major architectures currently being used are Command and Control architecture popularly abbreviated as C&C architecture and peer to peer architecture or P2P architecture. Peer to peer architecture was designed to overcome the shortcomings of the command and control architecture, it is an improvised version of the command and control architecture which will be justified shortly.

A. Command and control architecture(C&C)

The command and control architecture as the name suggests consists of controlling the compromised machines by issuing commands and directives. The issuing of these commands and directives takes place through a "control server". This control server directly communicates with the Bot-master also known as the attacker who informs the control server about the commands that need to be executed by the bots or compromised machines.

After receiving malicious commands from the attacker, the control server forwards these commands to the bots. These bots execute malicious commands on a target organization or an individual. The multiple compromised machines or bots in a botnet are collectively termed as the bot-army or Zombies. The control server is a kind of centralized control of all the bots. They install keyloggers that are intended to steal personal information such as credit-card details. They can send spam emails or help in conducting DDoS attacks.[9]

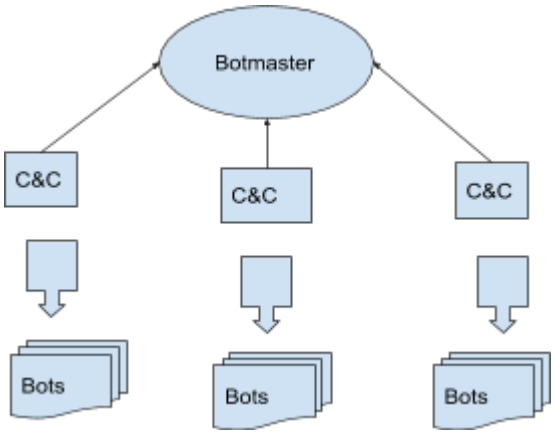

Figure 1 Command and Control Architecture

B. Peer to peer connection

The command and control architecture of a botnet exercises central control. If the defenders capture the control server, they obtain information regarding the entire botnet, because command and control center is a single point of control. If this single point of control is captured, the entire botnet can be shut down. To overcome this vulnerability attackers came up with an improvised version of botnet i.e. peer to peer botnet. Such a botnet does not contain any single point of control. Bots in this type of architecture are divided into two groups: client bots and servent bots.

Servent bots are those which have static, non-private IP addresses and are accessible from the global internet. They behave both as server and client. Client bots are those which have private, dynamically allocated addresses and they do not accept incoming connections. [9]

Only servent bots are candidates in peer lists. All bots, including both client bots and servent bots, actively contact the servent bots in their peer lists to retrieve commands. This design increases the network stability of a botnet. As the servent bots do not change their IP addresses. This bot architecture will hold more importance in the future as a larger proportion of computers will sit behind firewall, or use DHCP or private IP addresses due to shortage of IP space. Servent bots take the role of C&C servers: the number of C&C servers (servent bots) is greatly enlarged, and they interconnect with each other.

A peer to peer botnet does not receive commands from predefined places therefore it becomes necessary to perform authentication of commands issued. Authentication of commands require generation of a pair of private and public keys. The public key is hardcoded into the bot program before releasing and building the botnet. Hardcoding the key into the bot program eliminates the need for key distribution. To ensure the authentication and integrity of command messages, these are later digitally signed by the private key of the Bot-master.

In this botnet, each servent bot i, randomly generates its symmetric encryption key Ki. The peer list on bot A is denoted by PA. It will contain the IP addresses of M servent

IJSER

bots and the symmetric keys used by these servent bots. Thus, the peer list on bot A is

PA = { (IPi1, Ki1), (IPi2, Ki2), …...(IPiM, KiM)}

where (IPij, Kij) are the IP address and symmetric key used by servent bot ij. With such a peer list design, each servent bot uses its own symmetric key for incoming connections from any other bot. This is applicable because if bot B connects to a servant Bot A, Bot B must have (IPA, KA) in its peer list.

This individualized encryption ensures that if defenders capture one bot, they only obtain keys used by M servent bots in the captured bot's peer list. Therefore the encryption among the remaining botnet will not be compromised.

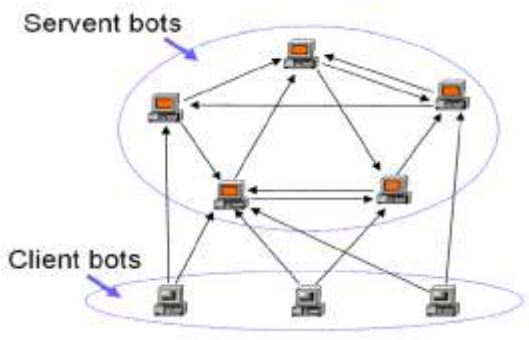

Figure 2 Peer to Peer connection

To monitor the peer to peer botnet, a special command called the report command is issued by the Bot-master which instructs every bot to send its information to a specified machine which is also compromised and controlled by the Bot-master .This data collection machine is called a sensor. The IP address of this centralized sensor host is specified in the report command. every report command issued by the Bot-master could utilize a different sensor host each time.

III. BOTNET CONSTRUCTION

Botnet connectivity is exclusively determined by the peer list in each bot. A natural way to generate peer lists is to construct them during propagation. To make sure that a constructed botnet is connected, the initial set of bots should contain some servent bots whose IP addresses are in the peer list on every initial bot.

A simple but effective worm can be used to construct a peer to peer botnet while it spreads from one host to another. Each worm infected computer has an "associate list" which contains the ip addresses of other infected hosts. As the worm spreads it constructs a peer to peer botnet structured according to this list.

This list can be built in the following way, when a host A infects another host B ,its associate list is passed to host B. Host A decides with a probability whether or not to replace one IP address in its own list with host B's IP address. If host B has already been infected before, host B updates a part of its own list with the new one sent from host A.

Other than ip addresses the associate list can also contain information supporting the propagation and maintenance of botnets. list on a host can contain the rough estimate of the bandwidth between this host and its neighboring bots. In this way, when the host needs to download the updated version of the bot program, it can select the host in the list with the highest bandwidth from which to download the update.

The botnet attacker can increase the controllability and response speed of his botnet by increasing the size of the associate list. We have introduced a means for attackers to construct a P2P botnet using a worm with associate list but it may also happen that a botnet encounters a honeypot while propagating. In order to detect and remove infected honeypots the worm is designed to have two parts. The first part compromises a vulnerable computer and then decides whether this newly infected machine is a honeypot or not; the second part contains the major payload and also the authorization component allowing the infected host to join in the constructed P2P botnet. Due to the different roles in a worm propagation, the first part is called the "spearhead", the second part the "main-force" of the worm.. The main force code lets the worm join the constructed botnet via the authorization key contained in the main-force.[10]

A worm which has two such parts is known as the two stage reconnaissance worm.one way to verify that a newly compromised host is a honeypot or not is to check whether the worm on it can infect other hosts on the internet.

The two stage reconnaissance worm's propagation procedure is illustrated by the following figure. The infection or propagation procedure begins by infecting a host B and checking whether it is a honeypot or not. The vulnerable host B is infected by the spearhead of the worm, which contains the exploit code and the associate list.

Next, the spearhead on host B keeps scanning the internet to find targets to infect them with the spearhead code. After the spearhead on host B successfully compromises n hosts including both vulnerable and already-infected ones, it tries to download the main-force of the worm from any host in its associate list that has the main-force component. [10]

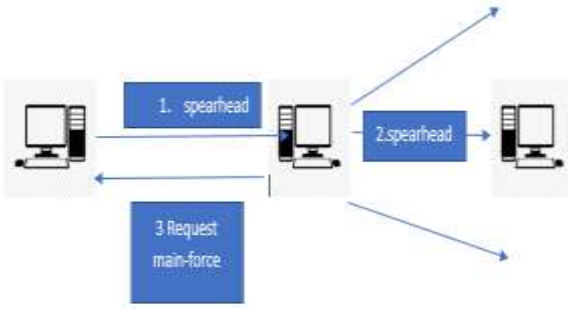

Figure 3

As a bot program propagates, the peer list in each bot is constructed according to the following procedures:

A. *New Infection*

Bot A passes its peer list to a vulnerable host B when compromising it. If B is a servant bot, A adds B into its peer list by randomly replacing one bot if its peer list is full. Similarly, if A is a servant bot, B adds A into its peer list in the same way.[9]

B. *Reinfection*

Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary

units (in parentheses). An exception would be the use of English units as identifiers in trade, such as "3.5-inch disk drive".[9]

IV. Botnet Detection

Since very little is known about the botnet malicious behavior. There are not many efficient techniques to detect a bot network architecture yet some recent studies prove to be effective. The study through honeynets helps much in understanding the working of the botnet architecture and the potent activities it can commit. This honeynets can also evolve into a detection mechanism. However it was meant for studying the technology and not necessarily detect a botnet infection [17].

The other approach is based on passive network traffic monitoring and analysis. Botnet detection techniques based on passive traffic monitoring have been useful to identify the existence of botnets. These techniques can be classified as being signature-based, anomaly-based, DNS-based, and mining-based that will be described and summarized in this section respectively.

A. Signature-based approach:

It was the first and most widely adopted method but not very successful. Its area of expertise is only limited to known botnets. Snort, an open source intrusion detection system (INS) [20][23], honey-nets and honeypots are systems used in monitoring network traffic to find the signs of intrusion. These are cost efficient while producing successful detection and analysis without false positives [21]. However, they are useless against unknown botnets unless they use techniques such as polymorphic and metamorphics [17] [20].

B. DNS-based approach:

Since bots are the ones who typically initiates the connection with the C&C server, for which they perform DNS queries to locate their server which is usually hosted on a Dynamic DNS provider (DDNS). Therefore, it is not difficult to detect a botnet by examining the DNS traffic and monitoring DNS traffic anomalies . In year 2005 a mechanism proposed by Dagon to identify C&C server by detecting abnormally high and concentrated DDNS query. But both the approaches generated many false positives by misclassifying legitimate DNSs with Short Time to Live (TTL).An another alternative approach was proposed in the following year based on abnormally recurring NXDOMAIN reply rates [24]. It was observed that name error NXDOMAIN on a DDNS often corresponds to a Botnet. They also suggested a set of technique based on heuristics that uses passive analysis of DNS based Black-hole list (DNSBL) lookup traffic to detect a botnet. With its help we can suspect and capture many domains names with less false positive.

Another promising technique came up with passive analysis of DNS based Black-hole list (DNSBL) lookup traffic, which helped in counter intelligence of inspection activity due to Supervisor-Bot ability to use DNSBL lookup before attack. However this has two problems; first having high false positive rate and second it cannot detect distributed inspection. In 2007 Hyunsang Choi et al. monitored group activities in DNS traffic generating group queries by distributed bots, which makes the same feature distinguished from normal legitimate DNS queries [25]. This approach can detect even encrypted channels along with C&C server migration as well More processing time required for monitoring, which is a problem with this approach.

Peter et al presented a system in 2009 which aim to detect Bot-infected machines, independent of any prior information about the C&C channels or propagation vectors, and without requiring multiple infections for correlation.

D. Anomaly-based Detection:

Anomaly-based detection techniques attempt to detect botnets based on several network traffic anomalies such as high network latency, high volumes of traffic, traffic on unusual ports, and unusual system behavior that could detect the presence of malicious bots in the network. Although anomaly detection techniques solve the problem of detecting unknown botnets, problems with anomaly detection can include detection of an IRC network that may be a botnet but has not been used yet for attacks, hence there are no anomalies [18][19]. To solve this, an effective algorithm that combines TCP-based anomaly detection with IRC tokenization and IRC message statistics to create a system that can clearly detect client botnets. This algorithm can also reveal bot servers [21].

However, this approach could be easily defeated by simply using a trivial cipher to encode the IRC commands. By this time an algorithm for detection and characterization of botnets using passive analysis based on flow data in transport layer got introduced. This algorithm can detect encrypted botnet communications. It helps to quantify the size of botnets, identify and characterize their activities without joining the botnet.

Recently, the Botsniffer that uses network-based anomaly detection to identify botnet C&C channels in a local area network [1]. Botsniffer is based on observation that bots within the same botnet will likely demonstrate very strong synchronization in their responses and activities. Hence, it employs several correlation analysis algorithms to detect spatial-temporal correlation in network traffic with a very low false positive rate [26].

V. Prevention

There is no hard way to prevent attacks from these botnets such as these but few primary secure ways can turn-out to create a safe environment so to list a few basic security measures can ultimately prevent more than a few network threats and hopefully our pc can remain safe:

A. Installing a windows firewall:

This is one of the most common and intuitive and preventive measures when it comes to averting botnet attacks. Though most users prefer disabling the firewall, a properly configured Windows firewall can block many network-based exploits. This measure is especially appropriate for large agencies with many similarly configured machines.[15]

IJSER

B. *Disable auto run:*

Auto run is a feature that allows operating systems to blindly launch commands from foreign sources. Doing so can bring into the system unwanted malicious files. Thus, it is always better to disable this feature.[15]

C. *Network-based intrusion prevention system (NIPS):*

These are designed to prevent a bot network attack from succeeding. A NIPS device is inserted in line with the traffic it is monitoring. Each packet is examined for an underlying known threat signature, if not found it is allowed to pass through the network. The disadvantage of this scheme is outdated signature matching lists.

D. *Network compartmentalization:*

In most computing environments, systems need not communicate with each other across departments. Shutting down this ability effectively works against the spread of botnets. Private virtual local networks and access control lists should be established by IT managers to limit network exposure as well as unauthorized access to resources.[15]

E. *Security Information Management Systems (SIM):*

A SIM system is a centralized database for network data. It collects, collates and organizes information so that information overload is not achieved on a network system. A SIM system performs data normalization also thereby making it easy for analyst to study the data. Information parameters are set up in a SIM to allow it filter out all irrelevant data it reaches. This not only relieves a network of excessive traffic but also improves network performance. SIM system reports are used in incident report management to aid in understanding network issues that arise. Bots increase network traffic and a SIM may pick up on this.

F. *Password protection in IoT devices:*

It is strongly recommended that IoT devices' passwords should be changed as soon as they are installed. Users of these devices often tend to use the default passwords, which makes them vulnerable to attacks. IoT devices are usually connected to the cloud and while choosing cloud providers, users should ensure that they have proper security mechanism in place to detect botnet attacks such as web application firewall and DDOS mitigation.

G. *Browser:*

It is advised to use a different browser for surfing on the mobile. The browsers for which most of the malware are written is on internet explorer and Mozilla Firefox. Scripts on mobile phones should be disabled in order to protect from attacks.

H. *Integrated security instead of security as an add on:*

IoT device designers, hardware manufacturers, software developers should try to introduce and integrate security when designing or developing their products rather than providing them as an additional feature when faced with a particular security issue.

VI. CONCLUSION

Botnets pose a significant threat to the security and infrastructure of a network and it continues to augment with the rapid increase in IOTs with peripheral security.
With limited research and outgrown network architecture of such network attack, preventive measures are not up to the mark. The sophistication and versatility pose the biggest threat the researchers [20] [22]. But Data mining-based techniques and DNS based botnet detection gives promising results creating more room for further research and development [23].

TABLE I. COMPARISON BETWEEN BOTNET DETECTION TECHNIQUES

	Unknown bot-detection	***Protocol & structure independent***	***Encrypted Based***	***Real-time detection***	***Low false positive***
Signature-based	No	No	No	No	No
Anomaly-based	Yes	No	Yes	No	No
DNS-based	Yes	No	Yes	No	Yes
Mining-based	Yes	Yes	Yes	No	No
Network based	Yes	Yes	Yes	No	No
Honeypot based	Yes	Yes	Yes	Yes	Yes

IJSER

REFERENCES

[1] G. Gu, J. Zhang, and W. Lee. "Bot Sniffer: Detecting botnet command and control channels in network traffic", In Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08), February 2008.
[2] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong, "Freenet: A distributed anonymous information storage and retrieval system", Lecture Notes in Computer Science, 2009.
[3] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang and D. Dagon, "Peer-to-peer botnets: Overview and case study", In USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07), 2007. [17] J. Goebel and T. Holz. Rishi, "Identify bot contaminated hosts by IRC nickname evaluation", In USENIX Wo rkshop on Hot Topics in Understanding Botnets (HotBots'07), 2007.
[4] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[5] K. Elissa, "Title of paper if known," unpublished.
[6] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
[7] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[8] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.
[9] Ping Wang ,sherri sparks,cliffc.zou,"an advanced hybrid peer to peer botnet"
[10] Cliff C. Zou ,Ryan "Cunningham.Honeypot-Aware Advanced Botnet Construction and Maintenance"
[11] Jai Puneet Singh,Akashdeep Chauhan, "Detection and prevention of non-pc botnets"
[12] What is mirai botnet, https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/

[13] The mirai botnet,https://www.csoonline.com/article/3258748/the-mirai-botnet-explained-how-teen-scammers-and-cctv-cameras-almost-brought-down-the-internet.html
[14] What is a botnet,https://www.veracode.com/security/botnet
[15] 11 ways to combat botnets https://www.esecurityplanet.com/trends/article.php/3920881/11-Ways-to-Combat-Botnets-the-Invisible-Threat.htm
[16] What is a botnet,https://www.iovation.com/topics/botnet
[17] Honeynet Project and Research Alliance "Know your enemy: Tracking Botnets, March 2005". http://www.honeynet.org/papers/bots/
[18] J.R. Binkley and S.Singh, "An algorithm for anomaly-based botnet detection," in Proc. USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI'06), 2006, pp 43–48
[19] 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
[20] Yong Tang, Shigang Chen "Defending Against Internet Worms: A Signature-Based Approach", 0-7803-896&9/051 2005 IEEE.
[21] Jan Goebel, Thorsten Holz "Rishi: Identify Bot Contaminated Hosts by IRC Nickname Evaluation", HotBots'07 Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, ACM
[22] H. Choi, H. Lee, H. Lee, and H. Kim, "Botnet Detection by Monitoring Group Activities in DNS Traffic," in Proc. 7th IEEE International Conference on Computer and Information Technology (CIT 2007), 2007, pp.715-720.
[23] Snort IDS web page. http://www.snort.org, March 2006.
[24] H. Choi, H. Lee, H. Lee, and H. Kim, "Botnet Detection by MonitoringGroup Activities in DNS Traffic," in *Proc. 7th IEEE International Conference on Computer and Information Technology (CIT 2007)*, 2007, pp.715-720.
[25] R.Villamarin-Salomon and J.C. Brustoloni, "Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic," in *Proc. 5th IEEE Consumer Communications and Networking Conference (CCNC 2008)*, 2008, pp. 476-481.
[26] M. M. Masud, T. Al-khateeb, L. Khan, B. Thuraisingham, K. W. Hamlen, " Flow-based identification of botnet traffic by mining multiple log file," in *Proc. International Conference on Distributed Frameworks & Applications (DFMA)*, Penang, Malaysia, 2008.

IJSER