

# The Survey Paper on ASP2P: An Advanced Botnet Based on Social Networks over Hybrid P2P

Saloni Shah<sup>1</sup>, Vina M. Lomte<sup>2</sup>

<sup>1</sup>Pune University, M.E., Department of Computer Engineering, RMD Sinhgad School of Engineering, Warje, Pune-58, India

<sup>2</sup>Pune University, Assistant Professor, Department of Computer Engineering, RMD Sinhgad School of Engineering, Warje, Pune-58, India

**Abstract:** *Botnets, regarded as the worms with fundamental command and control (C&C) channel, are becoming the typically frustrating threats to the Internet infrastructure. In order to defeat the botnets effectively, we should not only focus on known botnets, but also need to conduct some researches on the potential advanced botnets that could be developed by the adversary in the future. By investigating the vital C&C mechanism deeply, we present an advanced social peer-to-peer (ASP2P) botnet practically that combines the advantages of social networks and hybrid P2P structure in the way that bots receive encrypted commands from the social networks via HTTP and the C&C topology is hybrid peer-to-peer structure. ASP2P botnet is feasible and showed to be more covert and robust that it can be hardly detected and shut down compared with other P2P botnets.*

**Keywords:** botnet; social network; P2P; C&C

## 1. Introduction

According to explanation in [1, 2], malicious botnet is a network of compromised computers called "Bots" under the remote control of a human operator called "Botmaster". The term "Bot" is derived from the word "Robot"; and similar to robots, bots are designed to perform some predefined functions in automated way. In other words, the individual bots are software programs that run on a host computer allowing the botmaster to control host actions remotely [1, 2]. Botnets pose a significant and growing threat against cyber-security as they provide a distributed platform for many cyber-crimes such as Distributed Denial of Service (DDoS) attacks against critical targets, malware dissemination, phishing, and click fraud[3,4]. Botnet detection has been a major research topic in recent years. Researchers have proposed several approaches for botnet detection to combat botnet threat against cyber-security.

In this survey, botnet phenomenon will be clarified and advances in botnet detection techniques will be discussed. The botnets, in fact, have been moving to the web 2.0 such as online social networks to exploit the vulnerable social networks as the C&C servers [2, 3]. Therefore, the motivation of this paper is to construct an effective covert botnet with little effort to understand the current botnet sufficiently as well as defeat the potential next generation botnets. To achieve it, we proposed an advanced botnet named ASP2P botnet which utilizes the strengths of the social networks, which act as the C&C servers and hide the instructions, and hierarchical hybrid peer-to-peer structure to achieve more effective propagation and robustness. Meanwhile, the bots in the ASP2P botnet communicate via the widely used HTTP to enhance its imperceptibility.

The remainder of the paper is organized as follows: Section II describes botnet phenomenon. In this section, botnet characteristics and botnet life-cycle are explained to provide better understanding of botnet technology. Section III discusses botnet detection and tracking. In this section four classes of botnet detection approaches including signature based, anomaly-based, DNSbased, and mining-based are

discussed respectively. Section IV provides a brief comparison of botnet detection techniques. Section V provides the ASP2P botnet C&C mechanism. After that, concluding this paper in Section VI.

## 2. Botnet Phenomenon

Botnets are emerging as the most significant threat facing online ecosystems and computing assets. Malicious botnets are distributed computing platforms predominantly used for illegal activities such as launching Distributed Denial of Service (DDoS) attacks, sending spam, Trojan and phishing emails, illegally distributing pirated media and software, forced distribution, stealing information and computing resource, e-business extortion, performing click fraud, and identity theft[3,4].

The high light value of botnets is the ability to provide anonymity through the use of a multi-tier command and control (C&C) architecture. Moreover, the individual bots are not physically owned by the botmaster, and may be located in several locations spanning the globe. Differences in timezones, languages, and laws make it difficult to track malicious botnet activities across international boundaries [2, 5]. This characteristic makes botnet an attractive tool for cybercriminals, and in fact poses a great threat against cybersecurity. In order to provide better understanding of botnet phenomenon, botnet characteristics and botnet life-cycle will be explained respectively.

### 2.1. Botnet Characteristics

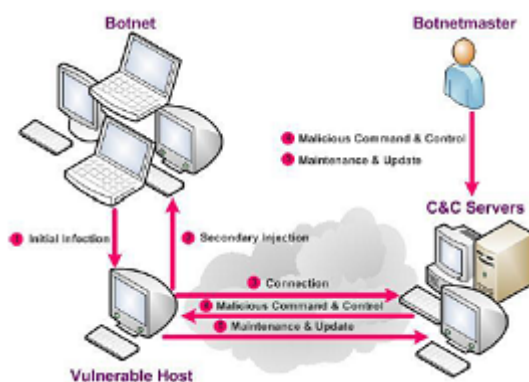
Like the previous generations of viruses and worms, a bot is a self-propagating application that infects vulnerable hosts through exploit activities to expand their reach. Bot infection methods are similar to other classes of malware that recruit vulnerable systems by exploiting software vulnerabilities, trojan insertion, as well as social engineering techniques leading to download malicious bot code [4,6,7]. According to measurement studies in [2] modern bots are equipped with several exploit vectors to improve opportunities for exploitation.

However, among the other classes of malware, the defining characteristic of botnets is the use of command and control (C&C) channels through which they can be updated and directed. The multi-tier C&C architecture of botnets provides anonymity for the botmaster. C&C channels can operate over a wide range of logical network topologies and use different communication protocols. Botnets are usually classified according to their command and control architecture [2, 4, 5, 6, 7].

According to their command and control architecture, botnets can be classified as IRC-based, HTTP-based, DNS-based or Peer to Peer (P2P) botnets [8]. P2P botnets use the recent P2P protocol to avoid single point of failure. Moreover, P2P botnets are harder to locate, shutdown, monitor, and hijack [9, 10]. However, according to the analysis in [2] the most prevalent botnets are based on Internet Relay Chat (IRC) protocol [11] with a centralized command and control mechanism. IRC protocol was originally designed for large social chat rooms to allow for several forms of communication and data dissemination among large number of end-hosts. The great prevalence of IRC-based botnets is due to the inherent flexibility and scalability of this protocol. Furthermore, there are several open-source implementations that enable botmasters to extend them according to their demands [2, 12].

## 2.2. Botnet Life-Cycle

A typical botnet can be created and maintained in five phases including: initial infection, secondary injection, connection, malicious command and control, update and maintenance. This life-cycle is depicted in Fig. 1.



**Figure 1:** A typical botnet Life-Cycle

During the initial infection phase, the attacker, scans a target subnet for known vulnerability, and infects victim machines through different exploitation methods. After initial infection, in secondary injection phase, the infected hosts execute a script known as shell-code. The shell-code fetches the image of the actual bot binary from the specific location via FTP, HTTP, or P2P. The bot binary installs itself on the target machine. Once the bot program is installed, the victim computer turns to a “Zombie” and runs the malicious code. The bot application starts automatically each time the zombie is rebooted [2, 8, 13].

In connection phase, the bot program establishes a command and control (C&C) channel, and connects the zombie to the

command and control (C&C) server. Upon the establishment of C&C channel, the zombie becomes a part of attacker’s botnet army. After connection phase, the actual botnet command and control activities will be started. The botmaster uses the C&C channel to disseminate commands to his bot army. Bot programs receive and execute commands sent by botmaster. The C&C channel enables the botmaster to remotely control the action of large number of bots to conduct various illicit activities [8, 13].

Last phase is to maintain bots lively and updated. In this phase, bots are commanded to download an updated binary. Bot controllers may need to update their botnets for several reasons. For instance, they may need to update the bot binary to evade detection techniques, or they may intend to add new functionality to their bot army. Moreover, sometimes the updated binary move the bots to a different C&C server. This process is called server migration and it is very useful for botmasters to keep their botnet alive [2, 8, 14, 15]. Botmasters try to keep their botnets invisible and portable by using Dynamic DNS (DDNS) [16] which is a resolution service that facilitates frequent updates and changes in server locations. In case authorities disrupt a C&C server at a certain IP address, the botmaster can easily set up another C&C server instance with the same name at a different IP address. IP address changes in C&C servers propagate almost immediately to bots due short time-to-live (TTL) values for the domain names set by DDNS providers. Consequently, bots will migrate to the new C&C server location and will stay alive [14, 15, 17].

## 3. Botnet Detection

Despite the long presence of malicious botnets, only few formal studies have examined the botnet problem. To date, just very little is known about botnet malicious behavior. The Honeynet project [4] was one of the pioneering informal studies of the botnet problem. However, efforts are in progress to quantify the botnet problem, detect the presence of botnets, and design defenses against attacks by botnets.

Botnet detection and tracking has been a major research topic in recent years. Different solutions have been proposed in academia. There are mainly two approaches for botnet detection and tracking [8]. One approach is based on setting up honeynets. For instance, solutions in [4, 18] have been initial honeynet-based solutions. In addition, many papers [2, 6, 7, 13, 19, 20, 21, 22, 23] discussed how to use honeynets for botnet tracking and measurement. However, honeynets are mostly useful to understand botnet technology and characteristics, but do not necessarily detect bot infection.

The other approach for botnet detection 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.

### 3.1. Signature-based Detection

Knowledge of useful signatures and behavior of existing botnets is useful for botnet detection. For example, Snort [24] is an open source intrusion detection system (IDS) that monitors network traffic to find signs of intrusion. Like most IDS systems, Snort is configured with a set of rules or signatures to log traffic which is deemed suspicious [24]. However, signature-based detection techniques can be used for detection of known botnets. Thus, this solution is not useful for unknown bots.

### 3.2. 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 indicate presence of malicious bots in the network [1].

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. To solve this, Binkley and Singh [25] proposed 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 [25]. However, Binkley's approach could be easily defeated by simply using a trivial cipher to encode the IRC commands.

In 2007, Karasaridis et al. [12] presented an algorithm for detection and characterization of botnets using passive analysis based on flow data in transport layer. This algorithm can detect encrypted botnet communications. It helps to quantify size of botnets, identify and characterize their activities without joining the botnet [12]. Recently, Guet et al. have proposed Botsniffer [26] that uses network-based anomaly detection to identify botnet C&C channels in a local area network. 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].

### 3.3. DNS-based Detection

DNS-based detection techniques are based on particular DNS information generated by a botnet. DNS-based detection techniques are similar to anomaly detection techniques as similar anomaly detection algorithms are applied on DNS traffic. As mentioned in Section II, bots typically initiate connection with C&C server to get commands. In order to access the C&C server bots perform DNS queries to locate the respective C&C server that is typically hosted by a DDNS provider. Thus, it is possible to detect botnet DNS traffic by DNS monitoring and detect DNS traffic anomalies [15, 17].

In 2005, Dagon [27] proposed a mechanism to identify botnet C&C servers by detecting domain names with

abnormally high or temporally concentrated DDNS query rates. This technique is similar to the approach proposed by Kristoff [28] in 2004. However, both techniques have the same weakness and could easily be evaded by using faked DNS queries. Furthermore, according to the evaluation in [17], this technique generates many false positives due to misclassification of legitimate and popular domains that use DNS with short time-to-live (TTL).

An alternative approach was proposed by Schonewille and Van Helmond [29] in 2006. This approach was based on abnormally recurring NXDOMAIN reply rates. In order to classify anomalous reply rates, they use the algorithms similar to those Dagon used for classifying analogous query rates. According to their observation DDNS responses indicating name error (NXDOMAIN) often correspond to botnet C&C servers that have been shut down by authorities. Hosts that repeatedly issue such queries may be infected with a bot and they may have the vulnerability to enable similar infection. According to [17], this approach is very effective to detect several suspicious domain names and there may be less false positive because NXDOMAIN replies are more likely to refer to DDNS than to other names. Ramachandran *et al.* [30] proposed a set of techniques and heuristics to identify botnets using passive analysis of DNS-based Black-hole List (DNSBL) lookup traffic. This technique addresses the possibility of performing counter-intelligence that help us to detect DNSBL reconnaissance activity, whereby botmasters themselves must perform lookups against the DNSBL to determine their bots' blacklist status. The goal in developing these models and heuristics is to distinguish DNSBL queries issued by botmasters from legitimate DNSBL traffic to identify likely bots. These heuristics could be used to detect reconnaissance activities in real-time and allows for active countermeasures. As botmasters usually perform reconnaissance lookups prior to the use of bots in an attack, this DNSBL counter-intelligence can be used for early warning to boost responses. Moreover, this detection technique does not require direct communication with any component of the botnet, and does not disrupt the botnet's activity. They have presented the first study that uses direct analysis of DNSBL logs to infer other types of network behavior. However, this technique runs the risk of false positives due to active countermeasures such as reconnaissance poisoning. In addition, this approach cannot detect distributed reconnaissance.

### 3.4. Mining-based Detection

One effective technique for botnet detection is to identify botnet C&C traffic. However, botnet C&C traffic is difficult to detect. In fact, since botnets utilize normal protocols for C&C communications, the traffic is similar to normal traffic. Moreover, the C&C traffic is not high volume and does not cause high network latency. Therefore, anomaly-based techniques are not useful to identify botnet C&C traffic. Several data mining techniques including machine learning, classification, and clustering can be used efficiently to detect botnet C&C traffic.

Geobl and Holz [31] proposed Rishi in 2007. Rishi is mainly based on passive traffic monitoring for unusual or suspicious

IRC nicknames, IRC servers, and uncommon server ports. They use n-gram analysis and a scoring system to detect bots that use uncommon communication channels, which are commonly not detected by classical intrusion detection systems [31]. However, this approach is quite limited, in that IRC nickname can be changed to resemble normal host. In addition, this method cannot detect encrypted communication as well as non-IRC botnets.

In 2008, Strayer *et al.* [32] proposed a network-based solution using machine learning techniques for detecting botnet traffic. They showed that evidence of botnet command and control activity can be extracted from flow characteristic using passive traffic analysis. They adopt a two stage process which first distinguish IRC flows, and then identify botnet C&C traffic from normal IRC flows [32]. Although these techniques are effective to detect some botnets, they are specific to IRC-based botnets. Moreover, for accurate analysis and detection these techniques require access to payload content. Thus, it cannot detect encrypted C&C traffic.

#### 4. Comparison of Botnet Detection Techniques

This section provides a brief comparison of botnet detection techniques. We have compared botnet detection approaches based on key features including: ability to detect unknown bots, capability of botnet detection regardless of botnet

protocol and structure, and botnets with encrypted C&C channels, real-time detection, and accuracy. This comparison is summarized in Table 1.

As Shown in this table, signature-based techniques can only detect known botnets, whereas the other classes are able to detect unknown bots. However, there are few botnet detection techniques [15, 33, 34] that can detect botnet regardless of botnet protocol and structure. These techniques will be effective even though botmasters change their C&C communication protocol and structure. On the other hand, detection techniques that require access to C&C payloads [24, 25, 31, 32] are less effective as botmasters tend to use encrypted channels for C&C communications. Among all detection techniques, the only approach that allows real-time detection is a DNS-based detection which uses DNSBL counter-intelligence to detect reconnaissance in real-time. However, active countermeasures run the risk of false positives. The most recent botnet detection techniques [33, 34] based on data mining as well as DNS-based botnet detection approach in [15] provide promising tradeoff. These methods are independent of botnet protocol and structure. Moreover, they are effective to detect encrypted C&C botnet communication. In overall, these techniques can detect real-world botnets regardless of botnet protocol and structure with a very low false positive rate.

**Table 1:** Comparison of botnet detection technique

	Detection Approach	Unknown Bot Detection	Protocol & Structure Independent	Encrypted Bot Detection	Real-time Detection	Low False Positive
Signature-based	[24]	×	×	×	×	×
	[25]	√	×	×	×	×
Anomaly-based	[12]	√	×	√	×	√
	[26]	√	×	√	×	√
	[27]	√	×	√	×	×
DNS-based	[28]	√	×	√	×	×
	[29]	√	×	√	×	√
	[30]	√	×	√	√	×
	[15]	√	√	√	×	√
Mining-based	[31]	√	×	×	×	×
	[32]	√	×	×	×	×
	[33]	√	√	√	×	√
	[34]	√	√	√	×	√

#### 5. ASP2P Botnet C&C Mechanism

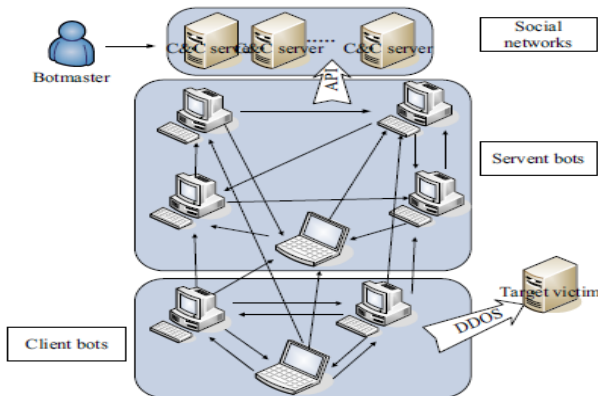
As stated before, the Command and Control mechanism is the significant part of the botnet. The major design challenge is to generate a botnet with a robust, covert and effective C&C mechanism that is difficult to be shut down, or monitored by defenders or other attackers. So, how to build a botnet with such kind of C&C mechanism is the big problem the adversary are facing now. However, we suppose the proposed ASP2P botnet use the following C&C mechanism to implement the robustness, covertness and effectiveness.

##### 5.1. Architecture of ASP2P Botnet

As illustrated in Fig. 2, the C&C channel is built like the hybrid peer-to-peer structure and the proposed ASP2P botnet can avoid, for example, the single point of failure. The bots are divided into two types: server bots and client bots. The server bots, acting as the server and client, receive the commands from the C&C servers or other server bots and forward the commands to the client bots as well as other server bots in the peer list. However, the client bots get the commands from the server bots in the peer list and execute the commands such as launching DDoS attack. As a consequent of it, the malicious instructions are propagated more quickly and make the botnet powerful in short time. In order to decrease the throughput generated by



the proposed botnet and hide the C&C channel as well as the encrypted messages effectively, we propose that the bots communicate with others or the C&C servers by using HTTP to transmit the covert information [6]. The firewall, in fact, is friendly to HTTP and the HTTP protocol is widely used in the world so that the covert messages could be drowned in the massive flows. Meanwhile, the malicious messages hidden in the HTTP protocol are encrypted completely so that it's not so easy for defenders to decrypt the information even if they have caught the malicious messages accidentally.



**Figure 2:** The proposed ASP2P Botnet. There're only three addresses of servent bots in the peer list of each client bot.

## 6. Conclusion

The botnets, indeed, have caused much damage to the Internet infrastructure and frustrated many people. We believe the botnet is evolving to be more covert and robust that is hard to be detected and defeated. In order to defeat the botnets effectively, it's necessary to analyze the current threat from the botnets we are facing today. However, it's equally important to conduct some researches on the potential advanced botnet that could be developed by the adversaries in the future. This paper presents an survey on advanced botnet named ASP2Pbotnet which exploits the merit of the social networks and combines the advantages of HTTP protocol and peer-to-peer structure. Compared with other P2P botnets, it is more difficult to be detected or monitored. It provides robust network connectivity, individualized encryption and covert communications. Simulation results shed light on the feasibility of the ASP2P botnet and show that the proposed botnet behaves secretly with low CPU usage, low memory consumption as well as low traffics and pretty good performances about the robustness and anti-detection. To defeat against such an advanced botnet, we consider that an anomaly-based detection may work. We should, therefore, conduct more researches to promote the detections to expose the potential botnets.

## References

- [1] B. Saha and A. Gairola, "Botnet: An overview," *CERT-In White Paper CIWP-2005-05*, 2005.
- [2] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in *Proc. 6th ACM SIGCOMM*

- Conference on Internet Measurement (IMC'06)*, 2006, pp. 41–52.
- [3] N. Ianneli, A. Hackworth, "Botnets as a vehicle for online crime," *CERT Request for Comments (RFC) 1700*, December 2005.
- [4] HoneyNet Project and Research Alliance. Know your enemy: Tracking Botnets, March 2005. See <http://www.honeynet.org/papers/bots/>.
- [5] G. Schaffer, "Worms and Viruses and Botnets, Oh My! : Rational Responses to Emerging Internet Threats", *IEEE Security & Privacy*, 2006.
- [6] 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.
- [7] A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers," in *Proc. ACM SIGCOMM*, 2006.
- [8] Z. Zhu, G. Lu, Y. Chen, Z. J. Fu, P. Roberts, K. Han, "Botnet Research Survey," in *Proc. 32nd Annual IEEE International Conference on Computer Software and Applications (COMPSAC '08)*, 2008, pp. 967–972.
- [9] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon, "Peer-to-peer botnets: Overview and case study," in *Proc. 1st Workshop on Hot Topics in understanding Botnets*, 2007.
- [10] P. Wang, S. Sparks, and C. C. Zou, "An advanced hybrid peer-to-peer botnet," in *Proc. 1st Workshop on Hot Topics in understanding Botnets*, 2007.
- [11] C. Kalt, "Internet Relay Chat: Client Protocol," *Request for Comments (RFC) 2812 (Informational)*, April 2000.
- [12] A. Karasiridis, B. Rexroad, and D. Hoefflin, "Wide-scale botnet detection and characterization," in *Proc. 1st Workshop on Hot Topics in Understanding Botnets*, 2007.
- [13] K. K. R. Choo, "Zombies and Botnets," Trends and issues in crime and criminal justice, no. 333, Australian Institute of Criminology, Canberra, March 2007.
- [14] 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.
- [15] 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.
- [16] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound, "Dynamic updates in the domain name system (dns update)," 1997. <http://www.faqs.org/rfcs/rfc2136.html/>.
- [17] 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.
- [18] N. Provos, "A virtual honeypot framework," in *Proc. 13th USENIX Security Symposium*, 2004, pp. 1–14.
- [19] M. Vrable, J. Ma, J. Chen, D. Moore, E. Vandekieft, A. C. Snoeren, G. M. Voelker, and S. Savage, "Scalability, Fidelity and Containment in the Potemkin Virtual Honeyfarm," in *Proc. ACM SIGOPS Operating System Review*, vol. 39(5), pp. 148–162, 2005.

- [20] F. Freiling, T. Holz, and G. Wicherski, "Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks," in *Proc. 10th European Symposium on Research in Computer Security (ESORICS)*, vol. Lecture Notes in Computer Science 3676, September 2005, pp. 319–335.
- [21] P. Barford and V. Yegneswaran, "An Inside Look at Botnets," ser. Advances in Information Security, Springer, 2006.
- [22] D. Dagon, C. Zou, and W. Lee, "Modeling botnet propagation using time zones," in *Proc. 13th Network and Distributed System Security Symposium (NDSS'06)*, 2006.
- [23] J. Oberheide, M. Karir, and Z.M. Mao, "Characterizing Dark DNS Behavior," in *Proc. 4th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2007.
- [24] Snort IDS web page. <http://www.snort.org>, March 2006.
- [25] 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.
- [26] 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.
- [27] D. Dagon, "Botnet Detection and Response, The Network is the Infection," in *OARC Workshop*, 2005.
- [28] J. Kristoff, "Botnets," in *32nd Meeting of the North American Network Operators Group*, 2004.
- [29] A. Schonewille and D.J. van Helmond. "The Domain Name Service as an IDS," Master's Project, University of Amsterdam, Netherlands, Feb 2006, <http://staff.science.uva.nl/~delaat/snb-2005-2006/p12/report.pdf>
- [30] N. F. A. Ramachandran and D. Dagon, "Revealing botnet membership using dnsbl counter-intelligence," in *Proc. 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI '06)*, 2006.
- [31] J. Goebel and T. Holz, "Rishi: Identify bot contaminated hosts by ircnickname evaluation," in *Proc. 1st Workshop on Hot Topics in Understanding Botnets*, 2007.
- [32] W. Strayer, D. Lapsley, B. Walsh, and C. Livadas, *Botnet Detection Based on Network Behavior*, ser. Advances in Information Security. Springer, 2008, PP. 1-24.
- [33] 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.
- [34] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol- and structure independent botnet detection," in *Proc. 17th USENIX Security Symposium*, 2008.

the M.B.A. degree in IT/Systems from Dnyangana Institute of Career Empowerment & Research, Pune University in 2012. Now studying in RMD Sinhgad College of engineering, Pune University for Post graduation in M.E.



**Vina M. Lomte** received the B.E. and M.E. Degree in Computer engineering. She is now working with RMDSSOE, Warje, Pune as Asst. Professor. She has experiences of 10 yrs 8 months and her Area of specialization - Web Security & S/W Engg.

## Author Profile



**Saloni Shah** received the B.E. degree in computer engineering from Cummins College of engineering for women, Pune University in 2009. And also received