ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

Detecting Peer-to-Peer Botnets using Conversation Tracking

Ruchi Dhole, Prof. Shobha Lolge

PG Student, Dept. of Computer Engineering, Lokmanya Tilak College of Engineering, Navi Mumbai, Mumbai University, Maharashtra, India

Assistant Professor, Dept. of Computer Engineering, Lokmanya Tilak College of Engineering, Navi Mumbai, Mumbai University, Maharashtra, India

ABSTRACT: Peer to Peer (P2P) systems are more vulnerable due to their open nature, also heavily targeted by malicious activities. The decentralized nature of Peer-to-Peer (P2P) botnets makes them difficult to detect. Their distributed nature also exhibits resilience against take-down attempts. Moreover, smarter bots are stealthy in their communication patterns, and avoid the standard discovery techniques which look for anomalous network or communication behavior. In this paper, we propose a technique to detect P2P botnet traffic and classify it from benign P2P traffic in a network. We use a 2-tuple 'conversation-based' approach which is port-oblivious, protocol-oblivious and does not require Deep Packet Inspection, instead of the traditional 5-tuple 'flow-based' detection approach. We are going to use classifier for getting better results at the end. The system proves higher performance, higher efficiency and lower maintenance cost, almost all malicious web pages are detected and the malicious codes encoded in the Csharp.

KEYWORDS: Botnet; Botnet Detection, Cyber-security, Command and Control Channel, Centralized, Decentralized ,Conversation Creation, Conversation Aggregation.

I. INTRODUCTION

The use of internet has tremendously increased all over the world. Web applications provide access to online services, gaining information from various sites and also a valuable target for security attacks. Gaining data from websites becomes more challenging in recent years due to the exploiting size of data, rising of dynamic web. Peer to peer (P2P) systems accomplish their tasks by collaborating number of peers into one network. Threat for security of P2P systems is ease of performing malicious activity and also not following the guidelines generated for secured communications. A Botnet can be considered as a network of bots under the remote command of a botmaster . In our previous survey paper [], we studied all the facts about the botnet as well as botnet life cycle. These bots are controlled to perform illicit activities. They pose a significant threat against cyber security. They provide a distributed platform for various cybercrimes such as distribute denial of service (DDOS), malware dissemination, click fraud and phishing. All users of computers are at high risk because we all browse the same internet. Every individual should be aware of social networking attacks.

Companies and governments suffer most damage from botnet attacks. The results of these attacks can be dangerous, costing the companies significant manpower, cost and clean. DDOS attacks can disrupt the communications and infected source code can halt the critical servers. Botnets have become much more sophisticated and dangerous now a day. Few formal studies have examined the botnet issues and very little is known about the malicious behavior of botnets. This research aims at finding out the latest and advanced techniques of botnet detection.

First of all, it is important to note that botnets should always be evaluated with metrics suited to the scope of the affected stakeholder groups.

The following examples illustrate the dependency on context when assessing the direct effects of botnets:

- Service providers who offer email services are interested in the amount of spam produced by botnets.
- Companies focusing on e-commerce may be primarily concerned about the power of DDoS attacks that can harm their ability to operate.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

- In order to protect their customers, financial organizations want to assess the potential of botnets for incurring financial loss.
- Governments need to shield themselves against the targeted theft of classified information.

The general goal of botnet measurement and threat characterization is to provide evidence that is helpful for:

- Deciding on investments in security technology and architecture. This is important for both governments and businesses.
- Defining the political agenda. The operation of botnets is major organized crime, and a threat to society, and therefore has to be engaged with at government level.
- Reporting and journalism. By providing information to the public, awareness of security issues and corresponding threats is increased.

II. RELATED WORK

As existing approaches to extract security dimensions suffer from scalability, it is working on static dataset. Traditional botnets were known to use IRC (Internet Relay Chat), which implied a centralized architecture for their 'Command & Control' (C&C) operations [1][2]. Detecting the centralized C&C server meant bringing down the entire botnet. Botmasters have utilized the resilience offered by P2P networks to build botnets wherein bots communicate, pass on commands and update other bots in a P2P fashion [5]. Just as a P2P network is resilient to break-down if a few peers leave the network, P2P botnets have proven to be highly resilient even if a certain number of bots are identified and taken-down. Previous P2P systems have many disadvantages like System not scale as the network size grows, work evaluates the detection of P2P botnets only with regular web [3][4]. This is a serious limitation because P2P botnet traffic (quite obviously) exhibits many similarities to benign P2P traffic. Distinguishing between hosts using regular P2P applications and hosts infected by a P2P botnet would be of great relevance to network administrators protecting their network. Most of the P2P systems using machine learning algorithm uses the approach is also limited to a binary classification [1][8].

Most networks use multiple firewalls and a layered security approach for protection against botnets[7]. Other steps that can be taken to prevent botnet attacks are: Full-Fledged Security Systems [9][11]: A lot of companies and organizations deploy full-fledged network security systems that cover all levels of the network from individual computers to the servers, local area networks, and external connectivity to the Web [14]. Another protection measure is shutting down unused ports that are not required for specific applications on the network. These are ports that are used for ftp applications and Internet Relay Chats which are the prime applications hackers use to get the bot computers to communicate with the bot herder[12][13]. Isolation involves putting a plan in place in the event of a botnet attack which isolates the infected computer from the network immediately after the attack is detected by the security system [16]. The infected computer is used to educate the organization on the security breach so a patch can be developed to repair the vulnerability. P2P traffic classification from the perspective of a more general problem of Internet traffic classification [8] or has given special attention to detection of botnets (centralized or distributed) in Internet traffic [17], [18], [19]. The detection of P2P botnet traffic in the presence of benign P2P traffic has not received much attention. Furthermore, the challenging context of correct categorization of the exact P2P application- whether benign or malicious- running on a host has received very little attention in past works [20], [21].

Initial work on detection of P2P botnets involved signature based and port-based approaches [22], which were easily defeated by bots which randomize their communication ports or use encryption. Although several approaches have been proposed to detect P2P botnets through the analysis of their network behaviour, most of them propose a binary classification of P2P hosts (i.e., benign or malicious) [18][23]. Some of the recent work has used supervised [7],[20] and unsupervised [24], [21], [8] machine learning approaches and other statistical measures [24], and have employed the standard 5-tuple categorization of network flows. Packets were classified as 'flows' based on the 5-tuple: <source IP, source port, destination IP, destination port, protocol>. Flows have bi-directional behaviour, and the direction of the flow is decided based on the direction in which the first packet is seen. This traditional definition of flows has been greatly employed and has seen huge success.

In response to this, a recent work [7],[19] has used the 2-tuple 'super-flows'(<source IP, destination IP>) with a graph-clustering technique to detect P2P botnet traffic. A graph-clustering approach may not scale as the network size grows. Further, their work evaluates the detection of P2P botnets only with regular web traffic (which was not analysed for the presence or absence of regular P2P traffic). This is a serious limitation because P2P botnet traffic (quite obviously) exhibits many similarities to benign P2P traffic, and distinguishing between hosts using regular P2P

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

applications and hosts infected by a P2P botnet would be of great relevance to network administrators protecting their network. Moreover, their approach is also limited to a binary classification.

III. PROPOSED ALGORITHM

A. *Steps to construct Botnet Tracking:*

Input: Network data, packets with two tuples.

Output: Labels of malicious nodes.

- Packet Sniffer captures all the packet that are travel through our network.
- All data logged to DB: All information of users gets recorded over log files and admin can collect it from server for further use.
- Admin look out log: The admin can get detail information about generated dataset and examines their behavior.
- Each entry matched and classification done
- The classification is done by using Bayesian algorithm and SVM algorithm and their results will be count on the basis of recent log files.

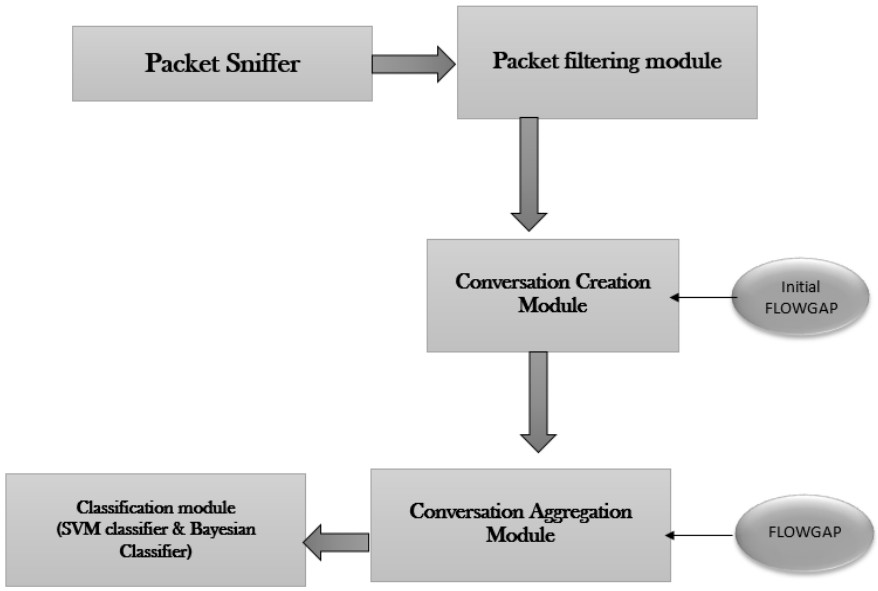

Fig 1: Botnet Detection Architecture using SVM and Bayesian Classifier

B. *Description of the Proposed Algorithm:*

Aim of the proposed algorithm is to maximize the network life by minimizing the total transmission energy using energy efficient routes to transmit the packet. The proposed algorithm is consists of three main steps. A 'conversation-based' detection mechanism which is protocol-oblivious, port-oblivious and payload oblivious, and relies only on the information obtained from the TCP/UDP/IP headers. Thus it does not require DPI, and cannot be evaded by payload encryption mechanisms. Detection of stealthy P2P botnet traffic inside a network, and differentiating it from regular P2P traffic. Categorization of the specific type of P2P application (regular or botnet) running on a host (with an accuracy of more than 95%).The five features used in this work are like:

- Current p2p traffic consideration during conversation.
- The duration of the conversation.
- The number of packets exchanged in the conversation.
- The volume of data exchanged in the conversation.
- The median value of the inter-arrival time of packets in that conversation.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

Module 1: Capturing live packets by Packet Sniffer:

In proposed system, there will be web server application. Web server will handle log of web server and log of network traffic. As we working on current traffic, we have to implement the sniffer to capture the live packet in the network may contain malicious packets as well as benign one. IP sniffing works through the network card by sniffing all of the information packets that correspond with the IP address filter. This allows the sniffer to capture all of the information packets for analysis and examination.Algorithm [1] specify our IP packet sniffer algorithm.

Algorithm [1]:

1. Initialize new instance of socket using specified address family, socket type and protocol.
2. Bind the socket to the selected IP address.
3. Set the socket options.
4. Sets low level operating modes for socket using IO Control Code enumeration to specify control codes.
5. Start receiving the packets asynchronously or iteratively from connected socket so that we can capture all incoming packets till stop the sniffer.
    a. Analyze the bytes received.
    b. Since all protocol packets are encapsulated in the IP datagram, so we start by parsing the IP header and see what protocol data is being carried by it.
    c. Now according to the protocol being carried by the IP datagram we parse the data field of the datagram.
    d. For each protocol type case: It finds the IP version, header length, differentiated services, total length, identification flags, Fragmentation Offset, time to live factor, protocol, checksum, source IP, destination IP.

Module 2: Packet Filtering Module:

This module takes in network logs in the form of raw packet data as input. The module reads each packet and isolates those which have a valid IPv4 header. From each packet, the Source IP, Destination IP, Payload length and Timestamp are extracted and stored for future use. For the purpose of data sanitization, all packets without a valid IPv4 header are deemed invalid and discarded. The packets are further filtered to keep only those packets which have a valid TCP or UDP header and a non-zero payload.

Module 3: Conversation Creation Module:

The output of the Packet Filtering module is fed as input to the conversation creation module. This module creates a list of conversations by aggregating packets received from the previous module. Each conversation is identified by the binary tuple <SIP, DIP> and an initial FLOWGAP value. The initial FLOWGAP is used to create conversations: while iterating through packets, if a packet is encountered which belongs to the IP pair of the conversation and whose time-stamp lies within FLOWGAP time from the beginning or end of the conversation, the packet is added to the conversation and the attributes of the conversation are modified accordingly [7].

Module 4: Conversation Aggregation Module:

The conversations created in the creation module are aggregated for a higher FLOWGAP value as desired by a network administrator. Here, the network administrator is given the flexibility to mine data for the time-period desired by him and giving him visibility into the network logs as desired by him. Such flexibility is especially valuable for bots which are extremely stealthy in their communication patterns and exchange as low as a few packets every few hours. For this evaluation, the value being used is 1 hour [7].

Module 5: Classification Module:

The Classification module uses supervised machine learning algorithms for training its model and classifying the test data. To validate our approach, models were built using a number of algorithms, namely Bayesian networks and SVM classifier. We get better results with the use of SVM over Bayesian.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

SVM are often considered as the classifier that makes the greatest accuracy outcomes in text classification issues as well as observed as cutting-edge models for binary classification of very high dimensional data. First, index the term in ascending order .Then, all the terms are weighted according to its features. If the score of weighting is greater than zero (weight>O), the term is classified as benign web page. Otherwise, the term is classified as malicious web page.

A support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. SVM classification algorithm is specified by Algorithm [2] as follows:

Algorithm[2]
Step 1: Start
Step 2: Read Log property
Step 3: Get list of Normal conversation
Get list of Malicious conversation
Step 4: Set bias, height, margin, hyperlane
margin = 1, bias = 0.5
Step 5: Classify
Set Training set and Testing set
NormalMalicious (Training set)
Goto step 6
Step 6: Classify hyperplane0
Get hyperplane property from log
If (hyperplane <=0.04)
Normal
Else
Malicious
Step 7: Find hyperplane
Hyperplane = margin/(weight*(∑ length property) + bias);
Step 8: Find NormalMalicious weight
Get OneClassLabel and ZeroClassLabel
Weight = ∑ (OneClassLabel) – ∑ (ZeroClassLabel)
Step 9: Show results
Step 10: End

IV. SIMULATION RESULTS

The simulation results involves the comparative study of the two classification algorithms i.e. Bayesian Classification Algorithm in table 1 and SVM Classification Algorithms in table 2. This comparative study is on the basis of Number of Normal conversation and Number of Malicious Conversation with respect to our Number of log files recorded in our dataset what we capture by our IP packet sniffer. The IP packet sniffer logs the packet into database and these logs are further given to classifier to classify them as normal or malicious. So we get the Number of Normal conversation and Number of Malicious Conversation.

Number of log files recorded	Number of Normal conversation	Number of Malicious Conversation
582	182	85
1050	198	134
2055	363	239
3060	528	344

Table 1 : Experimental results using Bayesian Classification Algorithm

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

Number of log files recorded	Number of Normal conversation	Number of Malicious Conversation
582	437	145
1050	850	200
2055	1735	320
3060	2620	440

Table 2 : Experimental results using SVM Classification Algorithm

The simulation results also involves the comparative study of the two classification algorithms i.e. Bayesian Classification Algorithm and SVM Classification Algorithms in table 3 on the basis of result count with respect to our Number of log files recorded in our dataset what we capture by our IP packet sniffer. The result count of both the classifier is in percentage format, which is with respect to table 1 and table 2. Also the Percentage Accuracy shows the accuracy of percentage of SVM over Bayesian Classifier.

Number of log files recorded	Result Count(%)		Percentage Accuracy (%)
	Bayesian Classification	SVM Classification	
582	45.87	95	49.13
1050	31.61	96	64.39
2055	29.38	95.5	66.12
3060	28.49	94.9	66.41

Table 3 : Comparison between Bayesian Classification Algorithm and SVM Classification Algorithms

The simulation results shows that Bayesian Classification Algorithm will fade away with number of log increases for every conversation as compare to SVM Classification Algorithms shows in fig 2 on the basis of different parameters like IP version, header length, differentiated services, total length, identification flags, Fragmentation Offset, time to live factor, protocol, checksum, source IP, destination IP, source port and destination port what we capture by our IP packet sniffer.

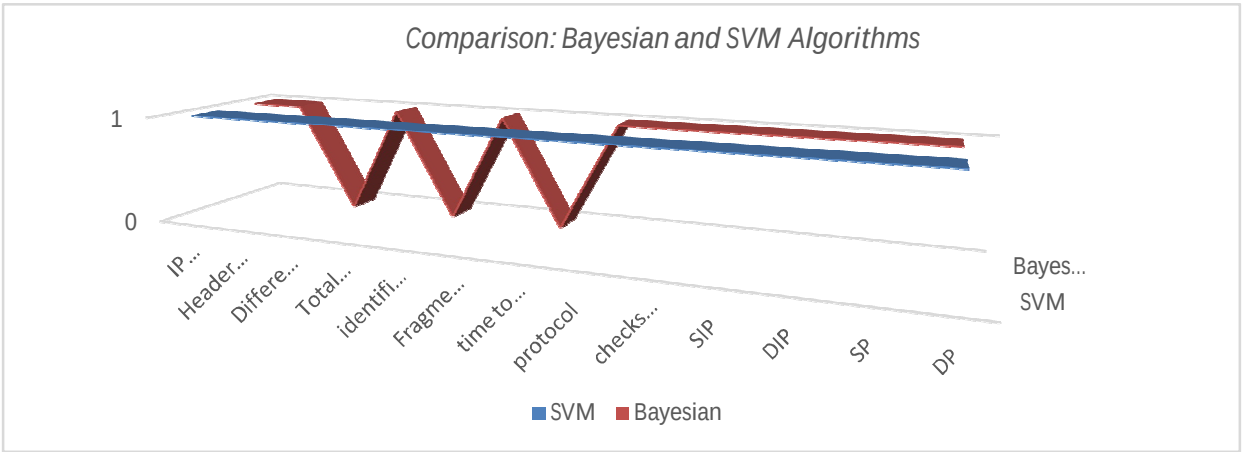

Fig 2 : Comparison between Bayesian Classification Algorithm and SVM Classification Algorithm on basis of different parameters

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

V. CONCLUSION AND FUTURE WORK

The existing system can work on live system as we develop our own IP packet sniffer which work good with respect to traffic. The results are calculated on the basis of Bayesian classification algorithm and SVM classification algorithm. Using threshold over the existing algorithm of SVM we found good results compare to basic algorithms. The experimentation results are quite outstanding in predicting the attacks. The identification of normal conversations and malicious conversations with respective parameters of classification algorithm is evaluated and analyzed. Now that we have developed a system that can predict multistage attacks, our aim is to improve the performance of the system, to add more features to find novel attacks.

REFERENCES

1. Piyush Chandekar and Dr. S. Chopra, "Efficient Sybil Attack Defence Mechanisms in Large Social Networks," International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Vol. 3, Issue 10, October 2015.
2. J.-T. Kim, H.-K. Park, and E.-H. Paik, "Security issues in peer-to-peer systems," in Advanced Communication Technology, ICACT
   a. pp. 1059–1063, 2005.
3. D. Dittrich and S. Dietrich, "P2P as botnet command and control: a deeper insight," in Malicious and Unwanted Software, MALWARE 3rd International Conference on. IEEE, pp. 41–48, 2008.
4. Felix Brezo, Jose Gaviria de la Puerta, Xabier Ugarte-Pedrero, Igor Santos, Pablo G. Bringas "A Supervised Classification Approach for Detecting Packets Originated in a HTTP-based Botnet" Clei Electronic Journal, Volume 16, Number 03, Paper 02, December 2013.
5. Daniel Plohmann, Elmar Gerhards, Felix Leder, "Botnets: Detection, Measurement, Disinfection & Defense" ENISA 2011.
6. Mohini N. Umale, Prof. A. B. Deshmukh, Prof. M. D. Tambakhe, "Review on Botnet Threat Detection in P2P" International Journal on Recent and Innovation Trends in Computing and Communication Volume: 3 Issue: 2, 2015.
7. Pratik Narang, Subhajit Ray, Chittaranjan Hota "PeerShark: Detecting Peer-to-Peer Botnets by Tracking Conversations" IEEE Security and Privacy Workshops 2014.
8. Ritu and Rishabh Kaushal "Machine Learning Approach for Botnet Detection", IEEE 2010.
9. David Barroso "Botnets – The Silent Threat", ENISA 2007.
10. GuofeiGu, Junjie Zhang, and Wenke Lee "BotSniffer: Detecting Botnet Command and Control Channelsin Network Traffic", IEEE 2008.
11. Guofei Gu1, Phillip Porras, Vinod Yegneswaran, Martin Fong and Wenke Lee "BotHunter: Detecting Malware Infection through IDSDrivenDialog Correlation" IEEE 2010.
12. Pijush Barthakur, Mrinal Kanti Ghoseand Manoj Dahal, "A Framework for P2P Botnet Detection Using SVM", International Conference on Cyber-Enabled Distributed Computing and Knowledge Discover 2012.
13. Elaheh Biglar Beigi, Hossein Hadian Jazi, Natalia Stakhanova and Ali A. Ghorbani "Towards Effective Feature Selection in Machine Learning-Based Botnet Detection Approaches", IEEE 2014.
14. Robert Walsh, David Lapsley, W. Timothy Strayer, "Using Machine Learning Techniques to Identify Botnet Traffic" Internetwork Research Department, BBN Technologies 2006.
15. Wu Xianghua, Ehsan Cao Lijun "Analysis and Design of Botnet Detection System", International Conference on Computer Science and Service System 2012.
16. M. Iliofotou, H.-c. Kim, M. Faloutsos, M. Mitzenmacher, P. Pappu, and G. Varghese, "Graph-based P2P traffic classification at the internet backbone," in INFOCOM Workshops, IEEE, pp.1–6, 2009.
17. G. Gu, R. Perdisci, J. Zhang, W. Lee et al., "Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection." in USENIX Security Symposium, pp. 139–154, 2008.
18. J. Franc¸ois, S. Wang, R. State, and T. Engel, "Bottrack: Tracking botnets using netflow and pagerank," in Proceedings of the 10th International IFIP TC 6 Conference on Networking - Volume Part I, pp. 1–14, 2011
19. H. Hang, X. Wei, M. Faloutsos, and T. Eliassi-Rad, "Entelecheia: Detecting P2P botnets in their waiting stage," in IFIP Networking Conference, pp. 1–9, 2013
20. B. Rahbarinia, R. Perdisci, A. Lanzi, and K. Li, "Peerrush: Mining for unwanted p2p traffic," in Detection of Intrusions and Malware, and Vulnerability Assessment, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, vol. 7967, pp. 62–82, 2013
21. J. Zhang, R. Perdisci, W. Lee, X. Luo, and U. Sarfraz, "Building a scalable system for stealthy P2P-botnet detection," Information Forensic sand Security, IEEE Transactions on, vol. 9, no. 1, pp. 27–38, 2014.
22. R. Schoof and R. Koning, "Detecting peer-to-peer botnets", University of Amsterdam, technical report, 2007.
23. S. Nagaraja, P. Mittal, C.-Y. Hong, M. Caesar, and N. Borisov, "Botgrep: Finding P2P bots with structured graph analysis." in USENIX Security Symposium, pp. 95–110, 2010.
24. J. Zhang, R. Perdisci, W. Lee, U. Sarfraz, and X. Luo, "Detecting stealthy P2P botnets using statistical traffic fingerprints," in Dependable Systems & Networks (DSN), IEEE/IFIP 41st International Conference, pp. 121–132, 2011.

BIOGRAPHY

Ruchi Dhole is pursuing M.E (Computer) from Lokmanya Tilak College of Engineering, Navi Mumbai, Mumbai University. She did his graduation B.E (Computer) from Mumbai University, Maharashtra. She is currently working on Conversation based detection of malicious peer-to-peer botnets.

Shobha Lolge is working as assistant professor in Dept. Of Computer Engineering at Lokmanya Tilak College of Engineering, Navi Mumbai, Mumbai University. She has academic experience of 12 years at UG and PG level courses of University of Mumbai. She has guided many projects at UG and PG level. Her area of interest are Database Management, Software Engineering, Mobile Computing, Artificial intelligence.