I.J. Computer Network and Information Security, **2014, 1, 48-54**
Published Online November 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2014.01.07

Auto-Pattern Programmable Kernel Filter (Auto-PPKF) for Suppression of Bot Generated Traffic

Kritika Govind
National Institute of Technology, Tiruchirappalli
Kritikagovind@yahoo.co.in

S. Selvakumar
National Institute of Technology, Tiruchirappalli
ssk@nitt.edu

Abstract—Bots usually vary from their other malicious counter parts by periodically reporting to the botmaster through regular exchange of messages. Our experiments on bot attack generation showed a continuous exchange of packets with similar content between the botmaster and the zombie machine at various time intervals. Though there were also genuine packets with similar content being sent out of the victim machine challenge was to differentiate between the two and pass only the genuine ones. In this paper, an algorithm namely Auto-Pattern Programmable Kernel Filter (Auto-PPKF), for automatic detection of patterns from packet payload for filtering out malicious packets generated by bots is proposed. The significant feature of our proposed Auto-PPKF algorithm is that, the malicious pattern is deduced at kernel level on the fly from packet payload. Traditional algorithms such as Boyer Moore, Knuth Morris Patt, and Naive Pattern search algorithms require the pattern to be identified available a priori. Currently, Longest Common Subsequence (LCS) algorithm stands as the most preferred algorithm for pattern matching. But the disadvantage is that common sequences can also exist in many genuine packets. Hence, the challenge lies in automatic detection of malicious patterns and filtering of the packets having such malicious patterns. This would not only put off the communication between the Botmaster and Zombie machine, but will also thus prevent user information from being sent to the botmaster.

Index Terms—Auto-PPKF, Bot, WFP, SpyEye Exploit Kit.

I. INTRODUCTION

Bot is a malicious piece of software derived from the word "ROBOT", which when installed in a host makes it a ZOMBIE machine. Botnets are a group of distributed bots controlled by a master computer often referred to as botmaster (attacker). Protocols such as Internet Relay Chat (IRC), Hypertext.

Transfer Protocol (HTTP), Peer to Peer (P2P) are exploited for the communication and propagation of bots. Bots usually communicate with their botmaster to receive commands from them. Based on this behavior, bots can be categorized into the following types:

IRC Bots: IRC, being a protocol designed for real time chat communication, has been exploited by attackers for achieving stealthy communication. In botnet scenario, the remote Botmaster in order to communicate with the zombie machines joins the specific IRC channel through a command and control structure. Detection of IRC bots is done at network level by observing various parameters such as High Traffic rate, large/Jumbo sized packets, long lived Communication using TCP ports, and by also looking in for commonly used IRC commands [1].

HTTP Bots: Bots have been designed to use HTTP for their communication. The advantage of using HTTP for implementing command and control is that almost all firewalls allow the HTTP traffic without blocking the port 80, since HTTP comprises a majority of Internet traffic. The detection of such kinds of bots is done by observing periodic repeatability in communication by checking the network traffic flow for long lived connections and then correlating the DNS queries for detection of abnormal communication pattern [2].

P2P Bots: Bots make use of the decentralized structure of the P2P protocol for communication. Initially infections are sent to victim as Trojan horse or through any other malicious programs. Then further secondary infections are spread to the Zombie machines through the P2P network. A case study on Peacomm bot shows that the infections are spread through the distributed hash table based on the kademlia algorithm. Detection of P2P bots is done based on the hash keys used [3]. Other existing solutions at kernel level are based upon filtering of packets using predefined signatures formed by analyzing the packet payload [4].

The rest of the paper is organized as follows. The related work carried out on bots is discussed under Section II. The Existing Detection techniques are discussed in Section III, followed by motivation for the proposed work in Section IV. Section V discusses the proposed solution along with the proposed algorithm

and its working. Discussions on the Experiments conducted viz., Bot Attack Generation, outcomes of the Experiments and performance analysis have been outlined under Section VI. Section VII includes the concluding remarks on the work done.

II. RELATED WORK

The traditional Pattern matching algorithms such as, Boyer Moore, Knuth Morris Patt, and Naïve pattern search algorithms require the pattern to be identified available a priori [5]. Rabin Karp string matching is based on hashing. But hashing at kernel level is an expensive operation [6]. The Hamming distance and levenshtein distance [7] between two strings are also currently used for pattern matching. Hamming distance requires that the strings to be compared should be of equal length and levenshtein distance results in approximate string matching. Longest Application Signature ExtRaction (LASER) algorithm exists for automatic signature generation of patterns using packet payload. But the disadvantage is that common sequences can also exist in many genuine packets which may lead to buffering of genuine packets resulting in increased overhead [8]. Honeycomb, which generates automatic signatures from packet payload uses Longest Common Substring (LCS) algorithm. Traffic is sorted on a per connection basis depending on the protocol traffic, port used, and then the signature is generated from the packet payload. Signature pool is maintained for comparing the signatures of the new packets against the existing signatures [9].

III. EXISTING TECHNIQUES

Botmaster makes use of the fast flux technique, viz., playing hide and seek with the Zombie machines, inorder to avoid the traceability of the origin of the attack. This fast flux technique refers to the rapid change in the IP address/domain name of the attacker. This change in identity of the attacker is further categorized into two types: Single flux and Double flux, wherein the attacker keeps changing the IP address with respect to one domain name and both the IP addresses and the domain name respectively. Existing detection techniques for such kinds of bots include Monitoring of Domain Name System (DNS) traffic [10], analyzing the network characteristics and performing data mining on them [11]. Identification of such kind of malicious bots in a network can be done by examining the pseudo randomly generated domain names [12]. Signatures of both the genuine programs and malicious programs are maintained as two separate lists namely white list and Black List. Agents are deployed for the detection of Botnets using Multi-Agent Technology [13] for monitoring traffic and the registry key changes. Based on these changes and correlation of connection details such as name of the process, protocols used, and ports used at a particular instant, the attack scenario is detected. Application specific malicious traffic identification [14] is also possible, where signatures of specific applications are obtained based on parameters such as packet payload, and packet flow information, viz., source address, destination address, protocol, port number, and inter arrival time of packets. Signatures are formed by taking a sequence of bytes from packet payload. Generation of datasets [15] and creating profiles for attack detection using machine learning techniques has gained momentum and is being used for detecting intrusions.

IV. MOTIVATION

The new algorithm for automatic detection of patterns and filtering of malicious packets at kernel level is an extension of our previous work on "Pattern Programmable Kernel Filter (PPKF)" for bot detection [16]. In PPKF, HTTP packets were checked for a predefined pattern which is known a priori at kernel level. In this paper an algorithm namely Auto-PPKF is proposed where the malicious pattern is deduced on the fly, by inspecting the packet payload at kernel level and then passed to PPKF for further processing. In a botnet scenario, the regular communication of similar/same data between the botmater and the Zombie machine to maintain its connectivity is considered to be malicious.

V. PROPOSED SOLUTION

In this paper a solution for automatic detection of patterns at kernel level has been proposed. Fig. 1, shows the block schematic of the proposed solution. In Fig. 1, notations A_1, A_2, … , A_n refer to the different applications in user space. The automatic pattern detector identifies the pattern by buffering the packets at kernel level to check for similarity among them. After deducing the pattern and confirming it to be malicious, they are updated in PPKF for comparing the forthcoming packets against them. If the incoming packets contain a pattern as in PPKF they are considered malicious and are blocked.

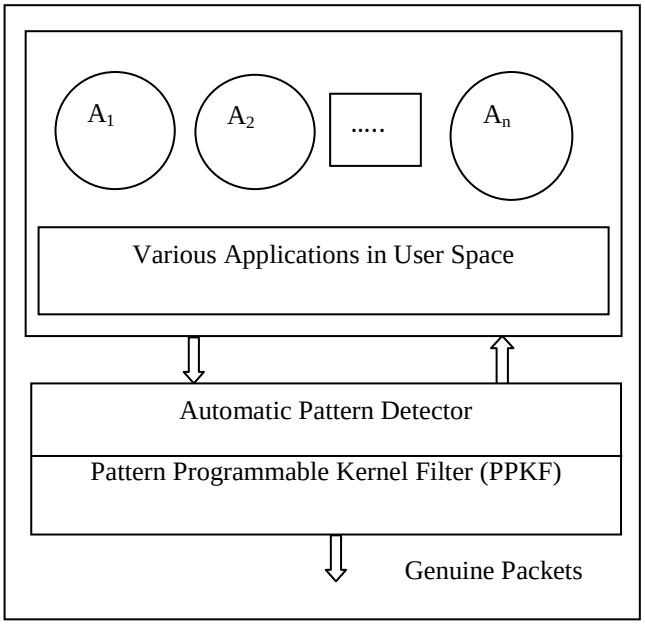

Figure 1: Block Schematic of the Proposed Solution for Bot Detection

A. Auto-PPKF Algorithm

- EXTRACT PATTERN
  - GET Next Packet
  - Check for Genuine Packet (Admin IP address)

  IF True
  - Genuine Packet
  - Allow Packet
  - IF End of Set COMPARE PATTERN
    Else GET NEXT PACKET

  ELSE Suspected Malicious
  - Extract Features
  - Update Pattern

  IF End of Set
  - COMPARE PATTERN

ELSE GET NEXT PACKET

- COMPARE PATTERN
  - GET Packets
  - Compare Packets

IF SIMILAR
- Update Common Pattern
- Block Packet

REPEAT: IF NOT End of List
- COMPARE PATTERN

ELSE IF NOT End of Set
- COMPARE PATTERN
  ELSE Allow Remaining packets
- EXTRACT PATTERN

ELSE REPEAT

B. Working of the proposed algorithm

The algorithm works on a set of N packets captured at kernel level. Initially a set of captured N packets are subjected to remote address verification. If the remote address corresponds to the Administrator IP address, then the packets are considered genuine and are allowed to the network. Else the packets are sent to the suspected malicious packet list (SM). Packet features such as Remote address, Packet length, and the Packet data are extracted from the packets in the SM list. Then the Compare Pattern module uses this set of N packets. The value of 'N' was set to thirty in our experiment which is a settable parameter. In Compare pattern module, the packets are checked for similar patterns. The packets are searched byte by byte in a brute force fashion to check if they contain similar pattern. If the pattern is found to be similar then, this pattern is taken as the malicious pattern generated by the bot and it is updated in the Common

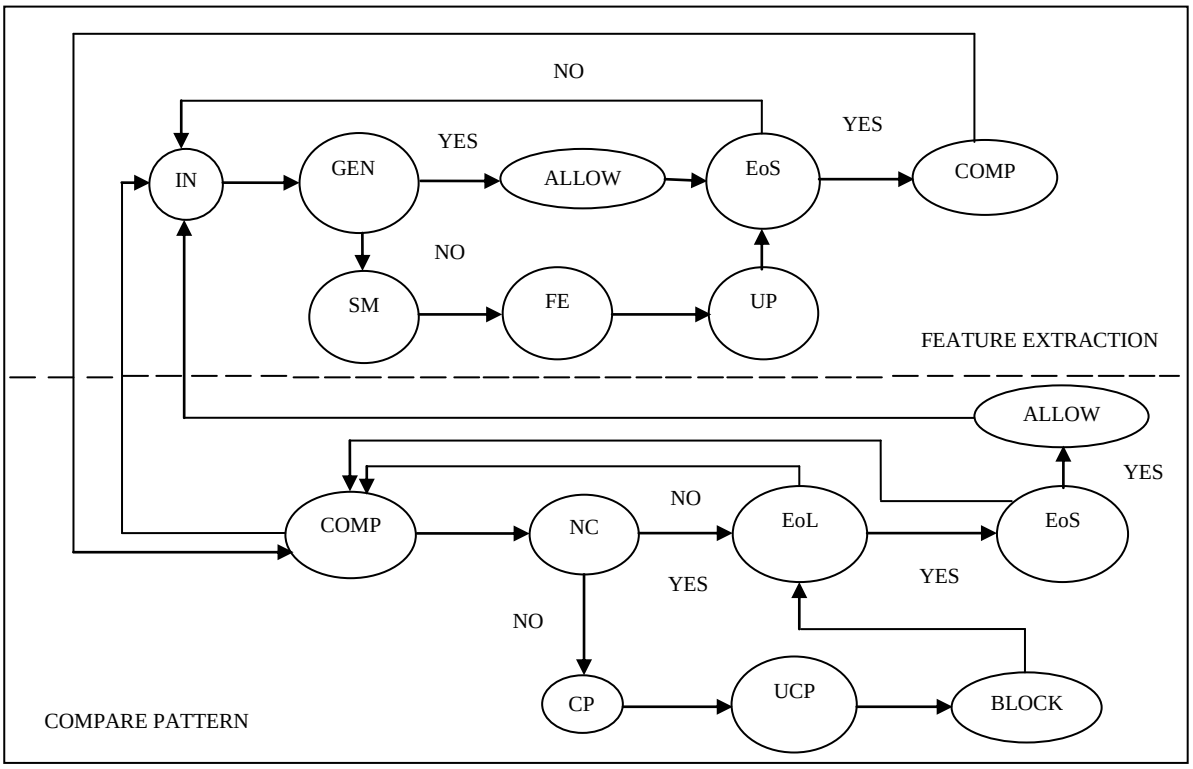

Figure 2: Finite State Automata for the proposed solution

IN: Initial State	GEN: Genuine Packet	ALLOW: Allow Packet	EoS: End of Set
COMP: Compare pattern Exists	SM: Suspicious Malicious	FE: Feature Extraction	UP: Update Pattern
NC: No Common Data Pattern	EoL: End of List patterns	CP: Common Data Pattern	UCP: Update Common
BLOCK: Block Packet			

Pattern (UCP) list. If there are no packets having common pattern and the comparison is over, the remaining genuine packets are allowed into the network. Then, the processing (checking for genuinity and feature Extraction) on next set of (N) packets commences.

VI. WORK DONE

A. Bot Attack Generation

Bot attack was generated with the Spy Eye Exploit kit. The connector interval time for the bot to report to its master was set as one minute and the malicious build file was generated, which was sent to the victim as an E-mail attachment. Traffic was captured using wireshark and continous exchanges of similar reporting packets between the botmaster and the Zombie machine were observed. Simultaneously genuine web traffic was generated with five tabs (five webpages containg text as well as pictures, viz., www.ndtv.com, www.yahoo.com, www.youtube. com, www.gmail.com, www.nitt.edu) open in the browser window and the traffic was captured for a period of half an hour.

B. Discussion on outcome of Experiments

The developed filter was loaded using a GUI namely Open System Resource (OSR). The filter can also be started and stopped using command prompt which may require making few registry changes manually. Changes being done to the existing WFP filter on loading the developed filter, were observed through event viewer. During the attack period a continuous exchange of 'HTTP GET' reporting packets between the bot master and zombie machine were noted. Genuine web traffic (five webpages containing text as well as pictures, viz., www.ndtv.com, www.yahoo.com, www.youtube. com, www.gmail.com, www.nitt.edu), as generated before loading the developed filter was generated and the traffic was captured using wireshark for the same period of half an hour. From the captured results it was found that the similar malicious HTTP GET packets between the botmaster and zombie machine were completely filtered allowing the rest of the genuine packets out of the network.

C. Performance Analysis

The performance of the developed Auto-PPKF was monitored using wireshark. Fig. 3 and Fig. 4 show the

Communication pattern between botmaster and Zombie machine before and after loading the filter respectively. The graph in Fig. 4 when compared to that in Fig. 3 shows the steep decline in communication pattern between the botmatser and Zombie machine after loading the filter. Fig. 5 and Fig. 6 show the HTTP Load Distribution between botmaster and Zombie machine before and after loading the filter respectively. From Fig. 6, it is seen that the HTTP load distribution between the botmaster and Zombie machine is Zero after loading the filter, thus suppressing the stolen user information from being posted on to the Botmaster.

VII. CONCLUSION

Bots being active malwares of today are propagating on every network used including the mobile network. So it is high time to come out with a solution to suppress the malicious communication of bots. Keeping this in mind an algorithm for automatic detection of malicious patterns from packet payload for filtering out malicious packets generated by bots at kernel level has been proposed, in this paper. The proposed solution was deployed in a bot attack scenario and it was found to work effectively. The proposed pattern matching algorithm can also be extended for the detection of other malicious programs such as worms which follow a particular communication pattern. The ongoing work is the identification of malicious process at host level and killing of the malicious bot process. Since SpyEye Exploit kit was recently released and not much research solutions were found addressing it, it was chosen as the demonstration environment for testing our proposed solutions for bot detection.

ACKNOWLEDGEMENTS

The authors thank the National Technical Research Organization (NTRO), New Delhi, Government of India for sponsoring this research work under the Collaborative Directed Basic Research in Smart and Secure Environment Project.

REFERENCES

[1] Claudio Mazzariello and Carlo Sansone.“Anomaly-Based Detection of IRC Botnets by Means of One-Class Support Vector Classifiers”. Proceedings of the 15th International Conference Image Analysis and Processing - ICIAP 2009, Vietri sul Mare, Italy, September 8-11, 2009. LNCS 5716, pp. 883– 892. Springer 2009, ISBN 978-3-642-04146-4_94.

[2] Jae-Seo Lee, HyunCheol Jeong, Jun-Hyung Park, Minsoo Kim, Bong-Nam Noh. “The Activity Analysis of Malicious HTTP-based Botnets using Degree of Periodic Repeatability”, International Conference on Security Technology, SECTECH’08, December 13-15, 2008, Hainan Island, China. Pages: 83-86. ISBN: 978-0-7695-3486-2.

[3] Yousof Al and Uwe Aickelin, “Behavioral Correlation for Detecting P2P Bots” Second International Conference on Future Networks, ICFN 2010. January 22-24, 2010, Sanya, Hainan, China. Pages: 323-327, ISBN: 978-0-7695-3940-9.

[4] SubhabrataSen, Oliver Spatscheck, Dongmei Wang, “Accurate, Scalable In-Network Identification of P2P Traffic Using Application Signatures”, *WWW2004*, May 17.22, 2004, New York, New York, USA. ACM 1-58113-844-X/04/0005.

[5] ‘Pattern Matching’ http://www.cs.princeton.edu/~rs/ AlgsDS07/21PatternMatching.pdf.

[6] Gaston H. Gonnet, Ricardo A. Baeza-Yates “An analysis of the Karp-Rabin String Matching Algorithm”, Information Processing Letters 34, 7 May 1990, North-Holland, pages 271-274.

[7] Jan Leeuwen, Handbook of Theoretical Computer Science: Algorithms and complexity, Volume 1, page no. 294, Elsevier Science, ISBN: 9780444880710.

[8] Byung-Chul Park, Young J. Won, Myung-Sup Kim, James W. Hong, “Towards Automated Application Signature Generation for Traffic Identification”, 2008 IEEE, ISSN no. 978-1-4244-2066-7.

[9] Christian Kreibich, Jon Crowcroft, “Honeycomb: Creating Intrusion Detection Signatures Using Honeypots”, ACM SIGCOMM Computer Communications Review, Volume 34, Issue Number 1: January 2004, Pages 51-56.

[10] Alper Caglayan, Mike Toothaker, Dan Drapaeau , Dustin Burke, Gerry Eaton, “Behavioral Analysis of Fast Flux Service Networks”, CSIIRW '09, April 13-15, Oak Ridge, Tennessee, USA , 2009. ACM 978-1-60558-518-5.

[11] AlperCaglayan, Mike Toothaker, Dan Drapeau, Dustin Burke and Gerry Eaton, “Real-Time Detection of Fast Flux Service Networks”, Proceedings of the 2009 Cybersecurity Applications & Technology for Homeland Security, CATCH’09. Pages 285-292, 2009 IEEE, ISBN No. 978-0-7695-3568-5.

[12] Peter Marko, Peter Vilhan, “Efficient Detection of Malicious Nodes Based on DNS and Statistical Methods”, SAMI 2012, 10th IEEE Jubilee International Symposium on Applied Machine Intelligence and Informatics, January 26-28, 2012, Herl’any, Slovakia.

[13] MiroslawSzymczyk, “Detecting Botnets in Computer Networks Using Multi-Agent Technology”, 2009 Fourth International Conference on Dependability of Computer Systems, ISSN No. 978-0-7695-3674-3, 2009 IEEE.

[14] Liang Lu, Jeffrey Horton, ReihanehSafavi-Naini, and Willy Susilo, "Transport Layer Identification of Skype Traffic" ICOIN 2007, LNCS 5200, pp. 465–481, 2008, Springer-Verlag Berlin Heidelberg 2008.

[15] Ali Shiravi, Hadi Shiravi, Mahbod Tavallaee, Ali A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection", Journal of Computers & Security 3I (2012), pp.357-374, Elseveir Publications.

[16] Kritika Govind, Vivek Kumar Pandey, and S. Selvakumar, "Pattern Programmable Kernel Filter for Bot dectection" Vol. 62, No. 3, May 2012, pp.174-179, Defence Science Journal (DSJ), DESIDOC India.

[17] Vadodil Joel Varghese and Stuart Walker, 'Dissecting Andro Malware', University of Essex 2011, Information Security Reading Room, SANS Institute.

[18] Experimental Security Analysis of a Modern Automobile, Karl Koscher, Alexei Czeskis, FranziskaRoesner, Shwetak Patel, and Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, HovavShacham, and Stefan Savage, 2010 IEEE Symposium on Security and Privacy, See http://www.autosec.org/ for more information.

SCREENSHOTS

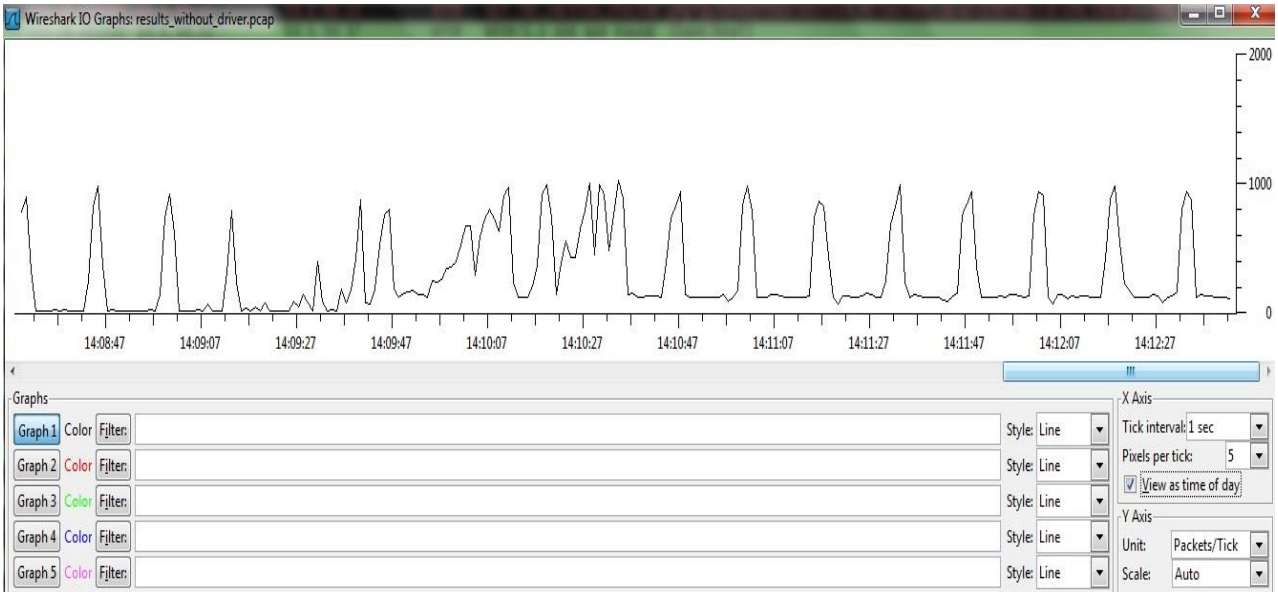

Figure 3: Communication Pattern between Botmaster and Zombie machine before loading the Filter

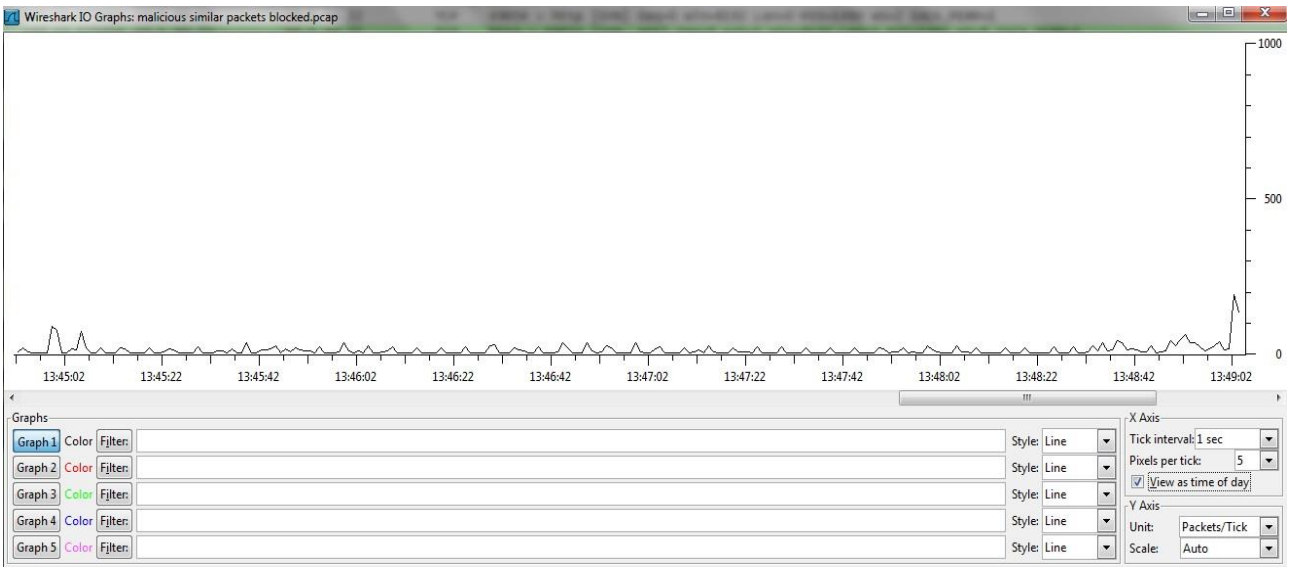

Figure 4: Communication Pattern between Botmaster and Zombie machine after loading the Filter

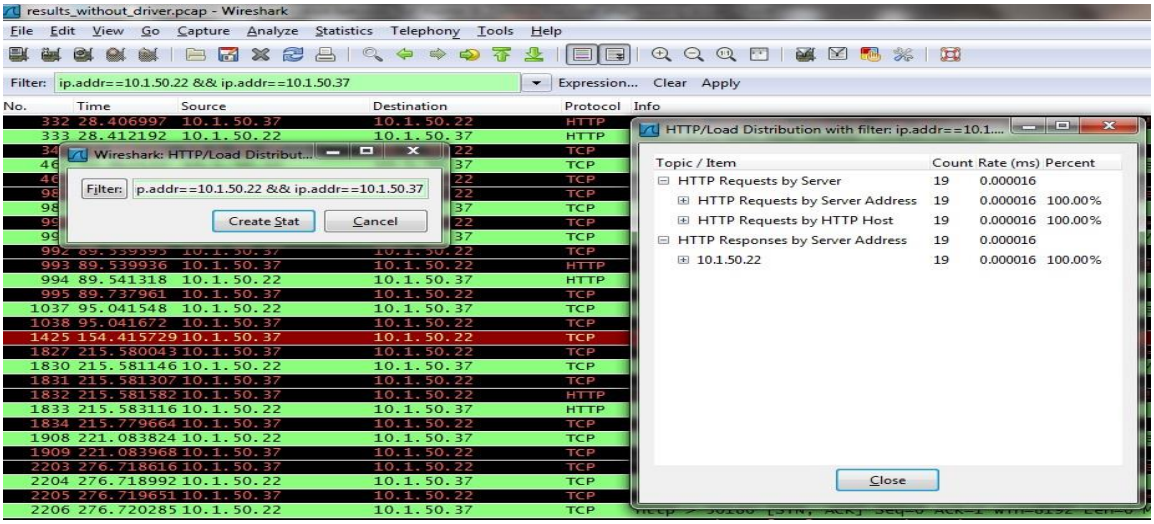

Figure 5: HTTP load distribution between Bot master & Zombie machine before loading Filter

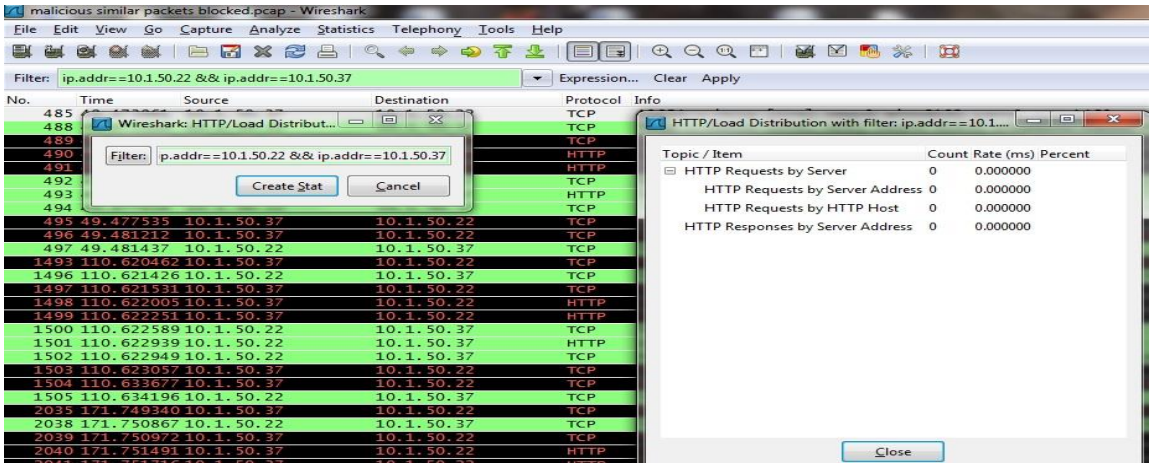

Figure 6: HTTP load distribution between Bot master & Zombie machine after loading Filter

Contributors

Ms. Kritika Govind received her B.E. (Computer Science and Eng.) from Sakthi Mariamman Engineering College, Anna University, Chennai, Tamil Nadu, in 2009. She completed her M. S. (by Research) in Computer Science and Engineering at National Institute of Technology, Tiruchirappalli, Tamil Nadu in 2013. Her areas of interest include Computer Networks and Cyber Security.

Dr S. Selvakumar is Professor and Head of the Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India. He got his B. E. Degree in Electronics and Communication Eng., from TCE Madurai (1983) and M. E. degree in CSE from REC, Tiruchirappalli (1987). He received his Ph. D. from the Indian Institute of Technology Madras (IITM), Chennai in 1999. His research interests include Network Security, Wireless Sensor Networks, Mobile Networks, Group communication in high-speed networks, Routing Protocols, Multimedia communication, and Scheduling for QoS guarantee. He has to his credit of publishing 62 research papers. Two Ph. D.s and two M.S. (by research) degrees have been awarded under his guidance and currently guiding three Ph.D. scholars and two M.S. (by research) scholars. He has completed a 5 year (2007-12) multi institutional research project on Collaborative Directed Basic Research in Smart and Secure Environment (CDBR-SSE) funded by NTRO, Govt. of India, New Delhi. He is presently the member of All India Board of IT Education, AICTE, New Delhi.

How to cite this paper: Kritika Govind, S. Selvakumar,"Auto-Pattern Programmable Kernel Filter (Auto-PPKF) for Suppression of Bot Generated Traffic", IJCNIS, vol.6, no.1, pp.48-54,2014. DOI: 10.5815/ijcnis.2013.01.07

