Optimized Solution for Denial-of-Service Attacks in Bloom-Filter-Based Forwarding

Anisha Sarah Mathew[1], Mitha Rachel Jose[2]

Department of Computer Science and Engineering, Mahatma Gandhi University, Mangalam College of Engineering, Kerala

Email id: anmathew00@gmail.com

9961617089

Abstract— Today, the Internet is an essential part of everyday life and many important and crucial services like banking, shopping, transport, health, and communication are partly or completely dependent on the Internet. Also it was originally designed for openness and scalability without much concern for security. Unfortunately, it is not possible to reliably determine the source of received IP packets, as the protocol does not provide authentication of the packet based on the source address field, can be easily pirated**.** Furthermore the Internet routing infrastructure also does not keep information about forwarded packets. Malicious users can exploit these design weaknesses of the internet to degrade its operation. Denial of Service attacks is launched by large number of compromised host to interrupt the services of legitimate users. Bloom Filter resolves current problem in the network such as routing table growth, multicast scalability and denial of service attack. Based on the information stored in the packet header as a filter, nodes will forward the packet. But it has a several problems like false positive issues, high memory and time usage. In order to overcome these problems, an algorithm called simulated annealing has been used, which results in the efficient detection of denial of service attacks in bloom filter forwarding.

Keywords— Bloom Filter, DoS, Network security, Simulated Annealing.

I. INTRODUCTION

Security is a fundamental component of every network design. A security policy defines what people can and can't do with network components and resources. Most security issues focus on the connection of the corporate network to the Internet and related issues such as viruses transmitted via electronic mail and intrusion from hackers. For secure communication to take place desirable security aspects such as confidentiality, authentication, message integrity and non repudiation is to be considered.

Classes of attack might include passive monitoring of communications, active network attacks, close-in attacks, exploitation by insiders and attacks through the service provider. A system must be able to limit damage and recover rapidly when attacks occur. Denial of Service is an attempt to flood an online service or computer resource by an attacker [13] with unwanted traffic in order to prevent it from functioning efficiently or reliably (Yuval et al., 2010). A number of sites are affected by such attacks , thus DoS attacks can be limited or completely eliminated by performing a change in packet forwarding logic in such a way that it will not affect Internet Protocol(IP) or other layers in TCP/IP protocol stack [12] . The Denial of Service attack incorporated multiple compromised machines and resources at a time to attack a sole target.

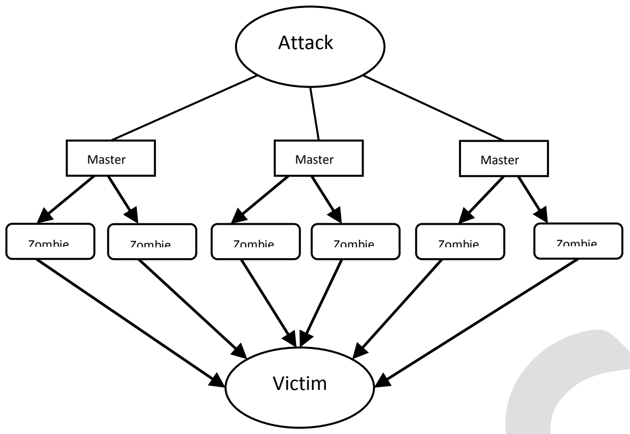

Fig. 1. Denial of Service Attack

A number of solutions are available as defence against the DoS attack however none can be a solution in standalone form and for longer time. Solutions include hop count filtering , Router base solution, Stack Pi, Push back router based defence , differential packet filtering , trace back through marking , trace back using entropy variations [4] and by using bloom filter technique. By using bloom filter technique [11] , computation capability of a machine enhances, access time of different files reduces. A Bloom Filter [4] is an ingenious randomized data structure for concisely representing a set in order to support approximate membership queries. The advantage of using bloom filter technique is as follows : Very little state and signalling required, native multicast support, no global addressing, path not revealed, no routing tables and lookups required.

In the existing system, it explains the intentional creation of routing loops and their anomalies by reverse engineering[1] the secret link identifiers with the help of a distributed adversary. It also explains how bloom filter forwarding are vulnerable to a form of packet injection that enables distributed flooding attacks by botnets [14] . It shows an attack against bloom filter discovery [5] in which a subscriber [15] prevents others from unsubscribing. The main drawback of the existing system is the rise of false positive result and absence of an accurate result.

In the proposed system, it defines that Bloom filter forwarding which prevents denial of service attacks in an optimized manner. This is done by using simulated annealing. Thus an optimized data analysis takes place. In the existing system, tree structure and connectivity data structures are used, instead of that, here graph is used. Hence time complexity and memory usage also reduces.

The rest of the paper is structured as follows : Section II gives an overview of the previous work on DoS preventive measures , Section III presents the Optimized solution for the DoS attacks in bloom filter forwarding , Section IV presents the experimental results and Section V concludes the paper.

II. Related work

Self-Routing Denial-of-Service Resistant Capabilities Using Inpacket Bloom Filters: Bloom-filter-based source-routing architecture which is resistant to Distributed Denial-of-Service attacks. This approach is based on forwarding identifiers that act simultaneously as path designators, i.e. define which path the packet [2] should take, and as capabilities, i.e. effectively allowing the forwarding nodes along the path to enforce a security policy where only explicitly authorized packets[10] are forwarded. The compact representation is based on a small Bloom filter whose candidate elements (i.e. link names) are dynamically computed at packet forwarding time using a loosely synchronized time-based shared secret and additional in-packet flow information (e.g., invariant packet contents).

Implementing zFilter based forwarding node on a NetFPGA: In this paper, it describes about the NetFPGA based forwarding node implementation for this new, IP-less, forwarding fabric. The implementation requires removing the traditional IP forwarding implementation, and replacing it with the Bloom-filter matching techniques for making the forwarding decisions. [6].

Secure in-packet Bloom Filter forwarding on the NetFPGA: In packet Bloom-Filter allows one to forward source-routed packets with minimal forwarding tables; the Bloom-Filter [1] is encoded with the identities of the links through which the packet needs to be forwardedif the link identities [7] are made content dependent. In this paper, it describes about the early implementation and testing of an in-packet Bloom filters forwarding node that implements cryptographically computed link identifiers.

Forwarding anomalies in Bloom filter based multicast: Several recently proposed multicast protocols use in-packet Bloom filters to encode multicast trees [5]. These mechanisms are highly scalable because no per-flow state is required in the routers and because routing decisions can be made efficiently by simply checking for the presence of outbound links [16] in the filter. This paper explores such anomalies, namely (1) packets storms, (2) forwarding loops and (3) flow duplication. This paper proposes stateless solution that increases the robustness and the scalability of Bloom filter-based multicast protocols [8] . In particular, it shows that the parameters of the filter need to be varied to guarantee the stability of the packet forwarding, and which presents a bit permutation technique that effectively prevents both accidental and maliciously created anomalies. The solution to avoid such anomalies [9] , the context of Bloom-Cast is proposed, which is a source-specific inter-domain multicast protocol, which uses analytical methods and simulations [3].

III. PROPOSED ALGORITHM

In this paper, we have presented a new solution for the DoS attacks in bloom filter forwarding. This done by using simulated annealing algorithm. Local Optimum is found each time and thus an accurate result is obtained with reduced false positive value . The system architecture is as shown below in Fig.2.

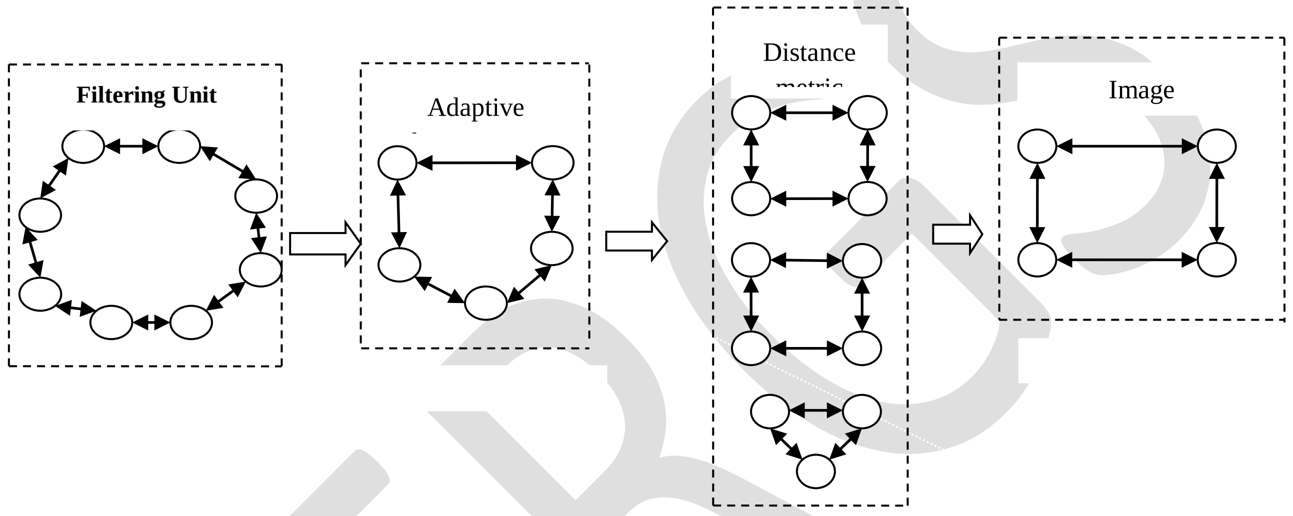

Fig.2.Detection of DoS using Simulated Annealing

Every incoming packet is received and it forms a graph; each node in the graph contains the sender, destination, data length and time details. Incoming packets are captured. The packets form graph, which is bidirectional, hence it takes less time and memory usage. A threshold is determined. Weight of each edge is found out. Highest weight of each node is found out. Thus the edges above the threshold value is considered . The different cycles among those edges are found out . From the above cycle obtained the attacker will be detected. For the result to be obtained in an optimized way , simulated annealing algorithm is used. Thus a local optimum is found out . An initial threshold value is set, based on that threshold value , local optimum is found out. Later access the threshold value for finding more optimized value until it reaches the stopping criteria.

In Optimized Bloom Filter Forwarding, there are mainly six phases: Packet Capturing Module, Attacker Module , Server Module, Filtering Module, Monitoring Module and Detection module.

A. *Packet Capturing Module*

The incoming packets are captured. These packets are filtered , monitored and detected to find whether it is an attacker group.

B. *Attacker Module*

Client machine acts as the attacker. Attacker attempts to make a machine or network resource unavailable to its intended users, such as to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet. Thus it results in complete shutdown of the target services.

C. Server Module

Server machine is the target machine to which packets are to be sent. Attacker tries to disrupt the network, preventing legitimate packets from getting through to their destination.

D. Filtering Module

The filtering unit captures the packet moving in and out of the network. Filtering unit is meant for providing protection against flooding attacks using real internet dataset.

E. *Monitoring Module*

The monitor's main task is to record the state (open, in progress, established) of any incoming session, information that will be utilized during the detection phase. Furthermore, it should be stressed that a single tracking memory solution can only be used as an indication of an attack, whereas the proposed monitor's information can be used, among others, for identifying malicious messages.

F. *Detection Module*

This unit is concerned with the detection of the threats or the malware. The detection is based on factors like number of port scans, unusually frequent logins and signups, high rate of packets send to a single port. There is a threshold associated with each of them described. This threshold value is automatically generated by applying a simulated annealing algorithm over the past history of traffic flows present in the database. The packet with fingerprints or signatures with values of the above given parameters exceeding their respective threshold will be classified as an attack. According to the unusual behavioural patterns like continuous port scans, unlimited connection request to the same IP or multiple packets destined to the same interface, extremely large number of packets with urgent pointer set, session hijacking etc, the attacks are detected.

Optimized Bloom Filter Forwarding consists of following steps:

1. Packet Capturing: Incoming packets are captured. These are represented as graph data structure , which is bidirectional hence time complexity reduces and memory usage reduces.

2.Abnormal node Calculation : Each node is considered , those node with edges more than 2 is selected. Those nodes are considered as attackers. This form an attacker group.

3.Detection of unusual cycles : To the attacker group , simulated annealing is applied and thus an optimized solution is obtained.

4.Detecting attacker groups : By applying simulated annealing False positiveness is limited with high accuracy.

While packet capturing takes place, packet analyzer intercepts and logs traffic passing over a digital network or part of a network. As data streams flow across the network, the sniffer captures each packet and, if needed, decodes the packet's raw data, showing the values of various fields in the packet, and analyzes its content according to the appropriate RFC or other specifications. False positiveness is the event of detecting a legitimate user or application as a threat. Increase in the false positiveness causes degradation in the overall performance factor of the detection system. By applying simulated annealing algorithm, the threshold value assigned such that all the packet flows exceeding that threshold value will be declared as an attack. This will reduce the false positiveness of the detection system when compared to the previous works.

IV. EXPERIMENTAL RESULTS

The performance of the proposed scheme is compare with the old scheme with respect to time complexity and memory usage. The Experiment is done on the basis of the memory usage when compared to our present system and the existing system. The figure 3 shows that memory usage decreases in our present system.

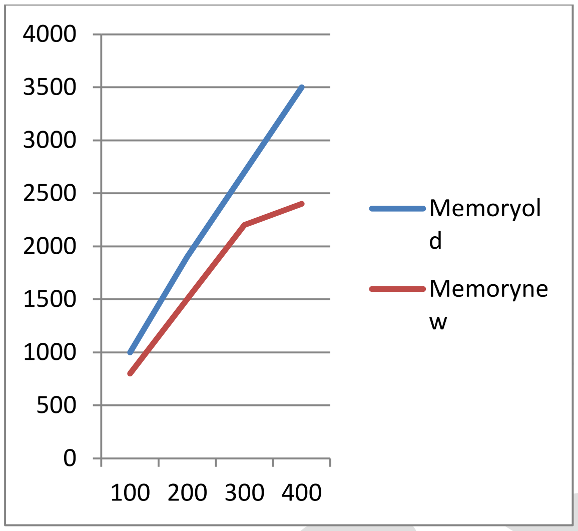

Fig 3 : Memory usage

The figure 4 shows that time taken decreases in our present system compared to the existing system due to the usage of graphs instead of trees.

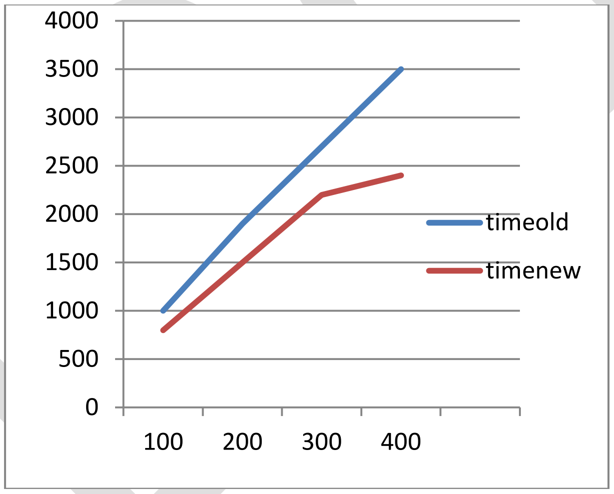

Fig 4: Time usage

V. Conclusion

DoS attacks can target tangible system resources, such as computational resources (bandwidth, disk space, processor time); configuration information (routing information, etc.); state information (for example, unsolicited TCP session resetting). This causes degradation in the overall performance factor of the detection system . The system presents how the denial of service attack is detected and prevented in bloom filter forwarding. In the present system , data analysis takes place using Simulated Annealing. Thus an optimized solution is obtained which result in less false positiveness and high accuracy.

ACKNOWLEDGMENT

I would like to acknowledge my project guide Ms Mitha Rachel Jose for her valuable support and useful guidance. I thank my friends and my family for being there with me.

REFERENCES:

[1] Markku Antikainen, Tuomas Aura, and Mikko Särelä, "Denial of Service Attacks in Bloom-Filter-Based Forwarding," IEEE/ACM Transactions on Networking, vol. 22, no. 5, October 2014

[2] S. Ratnasamy, A. Ermolinskiy, and S. Shenker, "Revisiting IP multicast,"*Comput. Commun. Rev.*, vol. 36, no. 4, pp. 15–26, 2006.

[3] C. Rothenberg, P. Jokela, P. Nikander, M. Särelä, and J. Ylitalo, "Self-routing denial-of-service resistant capabilities using in-packet Bloom filters," in *Proc. Eur. Conf. Comput. Netw. Defense*, 2009, pp. 46–51.

[4] P. Jokela, A. Zahemszky, C. Esteve Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: Line speed publish/subscribe inter-networking," in *Proc. ACM SIGCOMM*, 2009, pp. 195–206.

[5] M. Särelä, C. E. Rothenberg, T. Aura, A. Zahemszky, P. Nikander, and J. Ott, "Forwarding anomalies in Bloom filter basedmulticast," in *Proc. 30th IEEE INFOCOM*, 2011, pp. 2399–2407.

[6] C. Macapuna, C. Rothenberg, and M. Magalh aes, "In-packet Bloom filter based data center networking with distributed OpenFlow controllers,"in *Proc. IEEE GLOBECOM Workshops*, Dec. 2010, pp. 584–588.

[7] C. Rothenberg, C. Macapuna, F. Verdi, M. Magalhães, and A. Zahemszky, "Data center networking with in-packet Bloom filters," in *Proc. 28th SBRC*, Gramado, Brazil, 2010, pp. 553–566.

[8] A. Zahemszky, P. Jokela, M. Särelä, S. Ruponen, J. Kempf, and P. Nikander, "MPSS: Multiprotocol stateless switching," in *Proc. IEEE INFOCOM Workshops*, 2010, pp. 1–6.

[9] J. Keinänen, P. Jokela, and K. Slavov, "Implementing zFilter based forwarding node on a NetFPGA," in *Proc. NetFPGA Dev. Workshop*, 2009, pp. 1–8.

[10] A. Ghani and P. Nikander, "Secure in-packet Bloom filter forwarding on the NetFPGA," in *Proc. 1st Eur. NetFPGA Dev. Workshop*, 2010, pp. 1–7.

[11] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, pp. 422–426, Jul. 1970.

[12] M. Särelä, "BloomCasting for publish/subscribe networks," Ph.D. dissertation, Aalto Univ., Espoo, Finland, 2011.

[13] R. Lee, Z. Shi, and X. Yang, "Efficient permutation instructions for fast software cryptography," *IEEE Micro,*, vol. 21, no. 6, pp. 56–69, Nov.–Dec. 2002.

[14] T. Anderson, T. Roscoe, and D. Wetherall, "Preventing Internet denial- of-service with capabilities," *Comput. Commun. Rev.*, vol. 34, no. 1, pp. 39–44, 2004.

[15] D. Mazières, M. Kaminsky, M. F. Kaashoek, and E. Witchel, "Separating key management from file system security," *Oper. Syst. Rev.* vol. 33, pp. 124–139, Dec. 1999.

[16] D. Lagutin, "Securing the Internet with digital signatures," Ph.D. dissertation, School of Science and Technology, Aalto University, Espoo, Finland, 2010.