Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 12, December 2015, pg.74 – 78

An Optimized Review on Bloom-Filter Based Forwarding

M.Gurupriya[1], K.Pramilarani[2]

[1]Dept.of CSE & VTU, India

[2]Dept.of CSE & VTU, India

[1] priyamano89@gmail.com; [2] pramiselva@yahoo.co.in

Abstract - The fundamental problem in the network is rapid increase in routing table, scalability issues in multicast and Denial of Service attack caused by botnets. These problems have been overcome by using Bloom-Filter Based Forwarding. A Bloom filter is a data structure, used to test whether an element is a member of a set. The delivery tree is included in header of each packet. The nodes in the network with the help of bloom filter forward the packets based on their in-packet information without checking routing tables and without storing information in each and every node. But it has a several problems like false positive issues, time usage and high memory .To overcome the problems path selection algorithm is used.

Keywords- Multicast, botnets, delivery tree, path selection algorithm.

I.INTRODUCTION

Today security is the major concern in the Internet because the information present within the packets that are forwarded over the network are likely to be attacked by the attackers [2] .In Standard IP routing-table [1] if the user want to forward the data to any unicast or multicast groups the routing table is necessary to maintain in each node to store the flow and maintain state information .Hence it requires larger storage at each node in the network. When sender wants to send packets over a network without saving any information in the nodes, the best approach is Bloom-Filter based Forwarding. This technique is very simple .Each and every node has a delivery tree in its packet header as a set of Forwarding-hop identifiers (FHIDs)[3].

The set of FHIDs in the delivery tree constitutes a Bloom filter [3] data structure, which enables efficient testing of membership. Nodes in the network checks this potential FHIDs when they want to transfer the packets. Bloom filter provides better scalability when compared to standard IP routing because in latter state information is not stored. Denial-of-service attack is a method to make a destination node unavailable temporarily for users. There is high probability for this attack when forwarding data in the network, which leads to false positive or true negative

© 2015, IJCSMC All Rights Reserved

results. So that data forwarding chooses different path which is not in the routing table filter and it causes the data to loop inside the network itself.

To overcome DOS attack there are 3 security approaches that has been used in the bloom-filter structure
1) The number of items stored in bloom filter is limited.
2) Forwarding hop identifiers are kept secret by maintaining centralized bloom filter.
3) Cryptographically computed per-flow forwarding-hop identifiers.[1].

Bloom filter is a data structure for retaining data sets. The main functions of filter are adding the element and testing membership of the elements. Elimination is not possible in Bloom-Filter. The Bloom filter contains data element and the filter is built as a 8-bit array with constant length m, and k hash function is used in data set. k-bit is set to 1 when the result of testing is true. There are few chances that leads to false positive results during membership testing. The rate of false positive increases linearly with the number of n bits inside the filter. To reduce the false positive results limit the number of bits in the filter. The filters are designed in such a way that it fits inside network headers. The short filters has 256-1024 bits which can store nearly 20 to 100 data items. Bloom-filter follows three methods for multicast forwarding:

1) Bloom–filters are used in multicast to reduce space stored for forwarding routing table.

2) Multicast decision tree is encoded in packet headers.

3) Storing the lists of receivers in the packet header as a bloom-filter[4][5].

In the first method the outgoing interface of a multicast router has a Bloom-Filter which encodes the multicast-group addresses that are reachable through that interface. True negative are acceptable in the multicast to identify the packets which takes unnecessary paths. In the second method source tree or set of route through which data has to be processed is encoded in-packet Bloom-Filter. The outgoing link of each node is assigned with m bit link identifiers and k pseudo-randomly selected bits are set. In the third method the set of receiver through which data has to be sent is set inside the filter and when the data arrives each node will open the filter and checks whether it has any other node to forward. Based on the information saved in the filter forwarding decision will occur. But main drawback in this is security and possibility of denial of service attack which leads to false positive results.

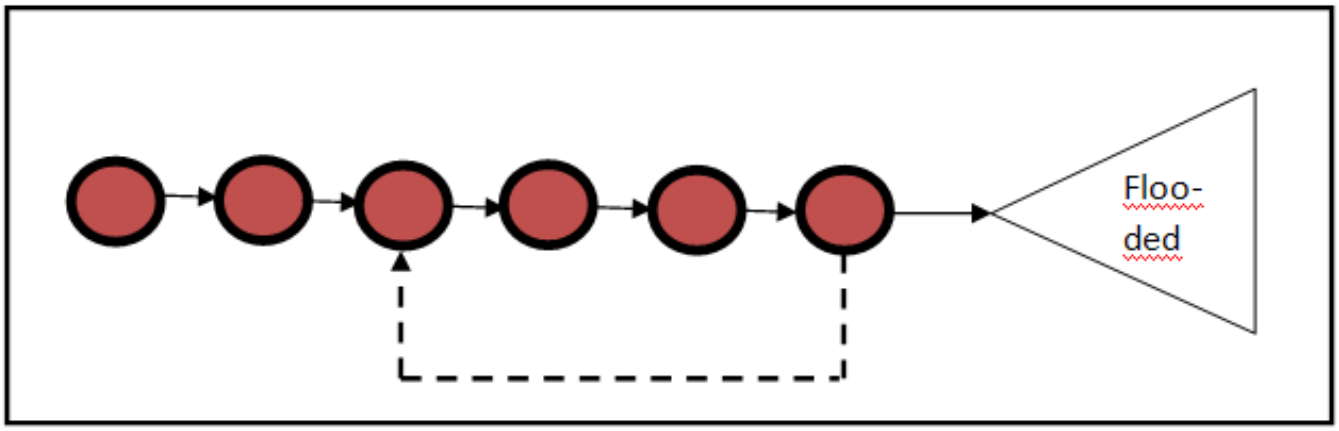

Fig 1: Forwarding Loop

© 2015, IJCSMC All Rights Reserved

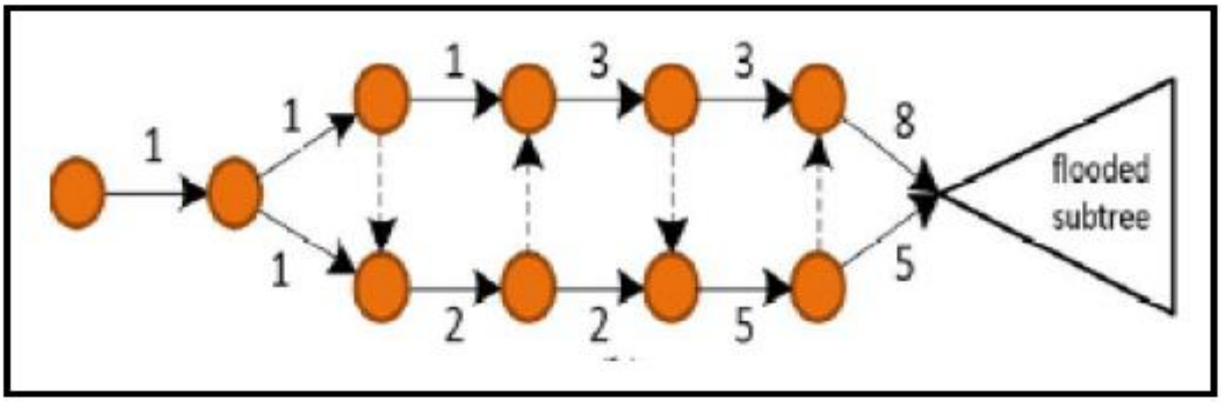

Fig 2: Repeated flow duplication[1].

II.RELATED WORK

A. In-Packet Bloom-Filters: This paper deals with an in-packet bloom-filter-based source-routing architecture, which is contrary to Distributed Denial-of-Service attacks. An in-packet bloom-filters is based on forwarding identifiers that act concurrently as path designators, i.e. represent which route the packet should take, and as potentiality, i.e. adequately allowing the forwarding nodes along the route to reinforce a security policy where only clearly authorized packets are forwarded. The close representation is based on a little bloom filter whose candidate components (i.e. Link names) are computed at packet forwarding time dynamically using a loosely integrated time-based mutual secret and additional in-packet flow information. The capabilities are thus usable and flow-dependent, but do not require any per-flow network state or memory lookup, preliminary security examination suggests that the self routing efficiency can be an effective building block towards Distributed Denial-of-Service resistant network architectures[8].

B. Revisiting IP Multicast: The paper revisiting IP multicast deals forwarding based on multicast. There are three different ways to send datagram to the destination in the network.one is unicast, second is broadcast and third is multicast. Unicast is one to one connection that is sending datagrams from one sender to one receiver. It consumes more bandwidth when audio and video are transferred. Broadcast means sending datagram from one node to all node in the network. The problem in broadcast is waste of bandwidth. Multicast send datagram to a particular group in the network, whose members are interested in the data. Multicast model has two advantages:(1)the effective use of bandwidth for communication.(2)Multicast is done with membership announcement. This paper proposes a new approach to implementing IP multicast that will leads to a re-evaluation of this commonly held view [9].

C. Self-Routing Denial-of-Service Resistant Capabilities Using Inpacket Bloom Filters[11]: This paper deals with in-packet Bloom-filter-based source-routing architecture which is resistant to Distributed Denial-of-Service attacks. This approach is based on forwarding data identifiers that act simultaneously as path designators, i.e. define which path the packet should take to reach destination, capabilities, i.e. Only secured and authorized packets are forwarded. The precise representation is based on a small Bloom filter whose data elements (i.e. link names) are computed dynamically at the time of forwarding packets using a loosely synchronized time-based shared secret and additional in-packet flow information. Any per-flow network state and memory look-ups are not required. Preliminary security analysis suggests that the self-routing capabilities can be an efficient building block towards DDoS-resistant network architectures [11].

D. Injection Attacks and Z-Formation: C. Rothenberg, P. Jokela, Nikander, M. Särelä, and J. Ylitalo [8], [1] identified possibility of injection attack in bloom-filter based forwarding and suggested that attackers can examine correlations among filters and therefore derived new filters allowed the injection attacks. Their planned solutions, called Z-formation calculate the link identifiers vigorously on per-packet basis. The bloom-filter, hash function is implemented as a cryptographic hash that accepts its parameters from a flow identifier in the packet, a sporadically

© 2015, IJCSMC All Rights Reserved

changing secret key local to each forwarding node, and the received and sent interface identifiers on the projected path of the packet. In this manner, the bloom-filters stores much more routing context as compared to the links on the routes. The need is to maximize the difficulty level of reverse-engineering the secret link identifiers or concatenating the filters of various flows (which either can be done by calculating their bitwise OR). Additionally, the path filters must be refreshed once in a while as the identifiers are time dependent, which will not allow the misbehaving senders to send to the path.

E. Secure in-packet Bloom Filter forwarding on the NetFPGA: In-packet bloom filter is a efficient way to forward packets based on forwarding table. Bloom filters contains the link information to reach destination .If the link information are dependent on previous node, e.g., by finding the next hop link over nodes. Bloom filters differ from packet – to –packet forwarding in unsecured traffic. This paper presents final implementation and testing of an in-packet bloom filters which works effectively on cryptographic identifiers. The performance and efficiency of algorithm is discussed in paper [12].

F. Data center networking with in-packet Bloom filters: This paper deals with best networking approach for cloud centered architecture based on the use of in-packet Bloom filters for identifying network approach. The major goals of cloud computing are scalability, cost, performance at many fields of data center environment. Motivated by the advent of high-radix, low-cost, commodity switches coupled with a content of programmability, our proposal contributes to the body of work re-thinking how to interconnect racks of commodity PCs at large. Rack Managers presented the flow based test –bed implementation for a data center architecture and supports true-negative forwarding and load balancing [13].

G .Bloom-Filter-Based Forwarding: Modern Internet suffers from variety of problems such as scalability problems, increase in routing table and Denial-of-Service attacks has been overcome by using bloom-filter based forwarding technique. The source node contains delivery tree which is embedded in packet header. Based on this information packets are forwarded to destination node without checking routing table and per flow information. The protocols have demanding vulnerabilities and make several false positive security guesses. Denial-of-Service attack against broad classes of bloom-filter-based protocols are presented and concluded that the protocols are not yet ready for deployment on open networks [15].

III.CONCLUSION

In this paper an optimized survey is made on bloom-filter based protocol to avoid denial-of-service attacks. The related works are In-Packet Bloom-Filters, Revisiting IP Multicast, Self-Routing Denial-of-Service Resistant Capabilities Using In packet Bloom, Injection Attacks and Z-Formation, Secure in-packet Bloom Filter forwarding on the NetFPGA, Data center networking with in-packet Bloom filters, Bloom-Filter-Based Forwarding. Our survey highlights the security analysis of bloom-filter based forwarding and their DOS vulnerability. The central conclusion is that bloom-filter based multicast is resistant to distributed packet flooding under very stringent assumptions.

ACKNOWLEDGEMENT

I am very thankful to my husband K.C.Rohit, IT Analyst, TCS for his cordial support, valuable information and guidance, to prepare this paper and also thankful to Prof. Dr. Prashanth C.S R, Head of the Department, Computer Science and Engineering, for his valuable and constructive suggestions during the planning and development of this work.

REFERENCES

[1] *Survey on Bloom-Filter Based Forwarding and Denial of Service Attack* IRoopa Patrimath, IINirmala Y Bariker
[2] *Survey On: Denial-of-Service Attacks with Bloom-Filters* Akshay Vinayak Parte, Anant Sunil Sonsale, Mayur Jagannath Dhole, Sharmila A. Chopade Computer Engineering, DYPIET, Ambi, Pune, India
[3] B. H. Bloom, *"Space/time trade-offs in hash coding with allowable errors*," Commun. ACM, vol. 13, pp. 422–426, Jul. 1970.

© 2015, IJCSMC All Rights Reserved

[4] B. Grönvall, "*Scalable multicast forwarding*," Comput. Commun. Rev.,vol. 32, pp. 68–68, January 2002.
[5] X. Tian, Y. Cheng, and B. Liu, "*Design of a scalable multicast scheme with an application-network cross-layer approach,"* IEEE Trans. Multimedia, vol. 11, no. 6, pp. 1160–1169, Oct. 2009.
[6] X. Tian, Y. Cheng, and X. Shen, "*DOM: A scalable multicast protocol for next-generation Internet,"* IEEE Netw, vol. 24, no. 4, pp. 45–51, Jul.–Aug. 2010.
[7] M. Särelä, C. E. Rothenberg, A. Zahemszky, P. Nikander, and J. Ott, *"BloomCasting: Security in Bloom filter based multicast,"* in Nordsec2010 Conference, 2011.
[8] P. Nikander, P. Jokela, C. Rothenberg, M. Särelä, and J. Ylitalo, "*Selfrouting denial-of-service resistant capabilities using in-packet bloomfilters*," in Proc. Eur. Conf. Comput. Netw. Defense, 2009, pp. 46–51.
[9] S. Ratnasamy, A. Ermolinskiy, and S. Shenker, "*Revisiting IP multicast,"* Comput. Commun. Rev., vol. 36, no. 4, pp. 15–26, 2006
[10] 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.
[11] 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.
[12] 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.
[13] 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.
[14] J. Keinänen, P. Jokela, and K.*"Implementing zFilter based forwarding node on a NetFPGA". in Proc. NetFPGA* Dev. Workshop, 2009, pp. 1–8.
[15] Markku Antikainen, Mikko Särelä , and Tuomas Aura, "*Denial-ofservice attacks in bloom-filter-based forwarding,"* IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 22, NO. 5, OCTOBER 2014.

© 2015, IJCSMC All Rights Reserved