Journal of Multidisciplinary Engineering Science Studies (JMESS)
ISSN: 2458-925X
Vol. 3 Issue 1, January - 2017

Detecting Ip Spoofing Using Hop Count Filtering Based Dynamic Path Update Approach

Srinath Doss
Fellow, Faculty of Computing
Botho University
Gaborone, Botswana
srinath.doss@bothouniversity.ac.bw

Sreekumar Narayanan
Fellow, Faculty of Computing
Botho University
Gaborone, Botswana
sreekumar.narayanan@bothouniversity.ac.bw

John Anand
Fellow, Faculty of Computing
Botho University
Gaborone, Botswana
john.anand@bothouniversity.ac.bw

***Abstract*—IP Spoofing has often been exploited by Distributed Denial of Service (DDoS) attacks to (1) conceal flooding sources and dilute localities in flooding traffic, and (2) coax legitimate hosts into becoming reflectors, redirecting and amplifying flooding traffic. Thus, the ability to filter spoofed IP packets near victim servers is essential to their own protection and prevention of becoming involuntary DoS reflectors. Although an attacker can forge any field in the IP header, he cannot falsify the number of hops an IP packet takes to reach its destination. More importantly, since the hop-count values are diverse, an attacker cannot randomly spoof IP addresses while maintaining consistent hop-counts. On the other hand, an Internet server can easily infer the hop-count information from the Time-to-Live (TTL) field of the IP header. Using IP to Hop Count mapping, the server can distinguish spoofed IP packets from legitimate ones. Based on this observation, we present a novel filtering technique, called Dynamic Path Update based HCF that builds Dynamic IP to Hope Count mapping table—to detect and discard spoofed IP packets. Dynamic Path Update based HCF is easy to deploy, as it does not require any support from the underlying network.**

***Keywords*— IP Spoofing, DDoS, HCF, DoS,Filtering**

I. INTRODUCTION

Network Security is one of the main domains of Information Technology and DDoS is one of the main threats to the Network Security. DDoS attacks servers and serves as hindrance to various security policies. The IP spoofing is one of the advanced methods of the DDoS attacks. Distributed Denial of services (DDoS) attacks is virulent, relatively new type of attack on the availability of internet service and resources [8]. DDoS attacker infiltrates large number of computers by exploiting software vulnerabilities, to setup DDoS attack networks. These unwitting computers are then invoking to wage a coordinator, large scale attack against one or more victims systems. As specific countermeasures are developed, attackers enhanced existing DDoS attack tools, deriving new techniques [9]. Hence, it would be desirable to develop comprehensive DDoS solution that defend against known and futures DDoS attacks variant. In 2000, there was severe attack on high profile website such has yahoo.com, CNN.com, amazon.com. In 2002, 8 out of 13 root DNS server were brought down has result of severe flooding denial of service attack [10]. Some proposal tries to detect spoofed senders using new routing mechanism such as “path markers supported by some or the entire router in root, as in Pi [11]. Few proposal try to detect spoofed senders using existing mechanisms, such as hop count (time to live)), as in Hop Count Filtering (HCF) [12]. However, empirical evaluation of these approaches show rather disappointing results [13].

The Source Router Preferential Dropping (SRPD) is proposed in [14]. The SRPD scheme monitors incoming high rate flows and preferentially dropped their packet. The dropping decision is based on flow rate threshold violation, the victim server’s response time, and the victim router queue occupancy. The IP trace back mechanism is one such approach to identify the hosts which were involved in an attack [15]–[19]. In a marking scheme [10], packets are marked probabilistically by intermediate routers, hence facilitating the victim network to identify the path traversed by the attack packets. A similar scheme, Tabu Marking Scheme (TMS) is proposed in [15][22].

(Hop Count Filtering (HCF) is one of the best methods to overcome IP spoofing. The IP protocol lacks the control to prevent a sender from hiding its packets’ origin. Moreover, routers don’t store IP address of each packet. Hence sender’s identity is not known by the routers. The TTL which is stored in each packet is accessed by the HCF so that it may be used to validate them and hence it does not need to trace back.

Based on hop-count, we propose a novel filtering technique, called Dynamic Path Update based Hop-Count Filtering (DPU based HCF), to weed out spoofed IP packets at the very beginning of network processing, thus effectively protecting victim servers’ resources from abuse. The rationale behind HCF is that most randomly-spoofed IP packets, when arriving at victims, do not carry hop-count values that are consistent with the IP addresses being spoofed. As a receiver, an Internet server can infer the hop-count information and check for consistency of source IP addresses.

In existing system (HCF), receiver check only unique path between source and destination that is specified in the IP2HC table. Hence the condition not satisfied, it will discard the packet. In our proposed system, dynamic path can be evaluated and updated with priorities in receiver table. During transmission time, if routers change the path, then the packet will be

discarded in receiver, according to the existing system. Using our proposed system, whenever the router chooses alternate path, the receiver checks each prioritized hop count only when hop counts is satisfied, then it will regard as legitimate packet else packet will be discarded.

II. Literature Review

During the earlier days, DDoS attacks were employed for IP spoofing. This was overcome by ingress filtering [1] that detected spoofed packet but it was not much effective. Reflectors [2] can be used to protect against the distributed denial of service attacks. Reflectors reflect the DDoS that have been sent by zombies or the hacker itself. These reflectors did not fulfill several hosts' expectation.

The study of previous proposals tries to detect spoofed senders using new rooting mechanism such as "path markers supported by some or the entire router in root, as in Pi [11]. The IP trace back mechanism is one such approach to identify the hosts which were involved in an attack [15]–[19]. In a marking scheme [10], packets are marked probabilistically by intermediate routers, hence facilitating the victim network to identify the path traversed by the attack packets.TCP service which has been used to great extent has been affected by IP spoofing. It is protected by various methodologies [3].

There is a scheme proposed for flooding attacks detection however, our research focus is on both high-rate and IP-spoofing attacks. The Multilevel Tree for Online Packet Statistics (MULTOPS) [20] provides a data structure for DDoS attack detection. The basic idea is that during normal operation, the packet rate of traffic in one direction is proportional to the packet rate in the other direction.

Jin et al. [21] proposed Hop-Count Filtering (RCF) for Internet servers to winnow away spoofed IP packets. The rationale behind RCF is that an attacker cannot alter the number of hops an IP packet takes to reach its destination, though he can forge any field in the IP header. The most randomly- spoofed IP packets, when arriving at victims, do not carry hop count values that are consistent with the IP addresses being spoofed. On the other hand, an Internet server can easily infer the hop count information from the TTL field of the IP header. Exploiting this observation, RCF builds an IP2RC mapping table to detect and discard spoofed IP packets, by clustering address prefixes based on hop counts[23].

Using a mapping between IP addresses and their hop-counts, the server can distinguish spoofed IP packets from legitimate ones. Based on this observation, we present a novel filtering technique, called Hop-Count Filtering (HCF)—which builds an accurate IP-to-hop-count (IP2HC) mapping table—to detect and discard spoofed IP packets. HCF is easy to deploy, as it does not require any support from the underlying network. Through analysis using network measurement data, we show that HCF can identify close to 90%of spoofed IP packets, and then discard them with little collateral damage. We implement and evaluate HCF in the Linux kernel, demonstrating its effectiveness with experimental measurements.

III. Proposed System

Dynamic Path Update based HCF which builds an all possibilities of IP-to-hop-count (IP2HC) mapping table—to detect and discard spoofed IP packets. DPU based HCF is easy to deploy, as it does not require any support from the underlying network. Through analysis using network measurement data, we show that DPU based HCF can identify more than 90% of spoofed IP packets, and then it check next possibilities (DYNAMIC) path to reach destination because there are many possibilities of routing path between source and destination. While the next path satisfies the condition then packet is forwarded to the receiver and update the HCF table else packet is discarded. In existing system (HCF), receiver check only accurate path between source and destination if it does not satisfied then packet is discarded.

In our system dynamic path can be evaluated and filled in receiver table. In case of alternate path during the traffic time, the alternate path were checked at the receiver table and packet is acceptable if it satisfies the condition else packet is discarded.

As per our proposed work, sender will have to first store the data in the sender buffer. Since attacker can easily evade the security barriers of the system, it also stores the data spoofing the sender's identity. So packet from the both, sender and attacker is attacked with experimental threshold (Te) and forwarded to the intermediate routers. It is then forwarded to receiver buffer. Here, each packet is separated into 3 fields. Data from packet is given to the Buffer. Actual TTL packet is extracted and forwarded to the DPU based HCF. The IP address from the packet is mapped with the IP2HC table to get the corresponding Hop Count (or Threshold) which has highest priority. When Te does not match with the corresponding Ta, then next highest priority Te is obtained. This is followed till the nth priority Te. The resultant status is given to the Buffer. The Buffer after analyzing the status accepts or discards the packet. This is shown in fig: 1. Sender should be initialized with the port number and IP address. The sender should be authenticated before allowing it to send the packet. Packet should be attached by sender along with its IP address and TTL field. With each packet the sender sends, the HCF obtained from its corresponding threshold time.

The packet along with its HCF (as given in the fig: 2) and IP field is fed into the sender buffer (as in fig: 3). Router receives each packet and forwards to the receiver host with minimal traffic life time. Each time the router receives the packet it attach its TTL field along with it. The router forwards each packet to the receiver system. Receiver stores each packet in its buffer (as given in fig: 4). It extracts the IP and TTL field and forwards the IP address to IP2HC table. The information obtained from this module is again forwarded to receiver.

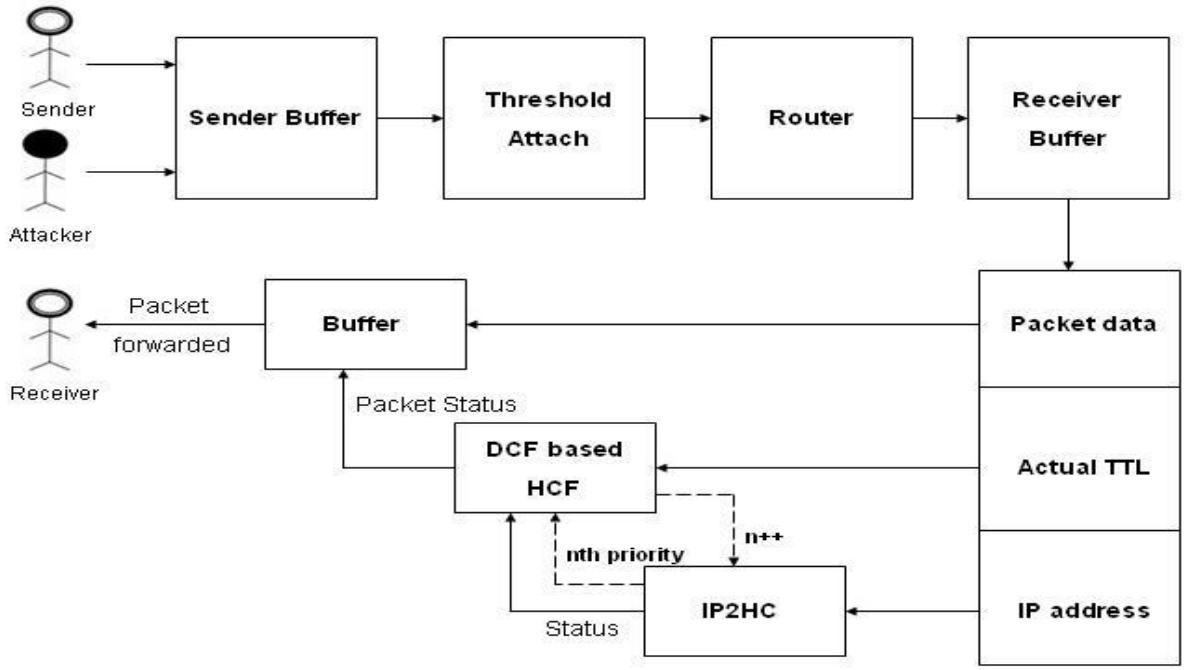

Fig 1: Overall System Design

Host name	TTL(Microseconds	*Priority*
System 1	3	1
System 2	300	1
System 2	900	2
System 2	1000	3
System 3	500	1
System 4	600	1
System 5	550	1
System 6	700	1
System 7	850	1
System 8	800	1
System 9	900	1

Fig 2: Initial HCF Table

Source IP	Protocol	*Packet*	*Status*	*Sender Time(micro Seconds)*
192.168.6.61	TCP	Hi	Forward	1335523149640
192.168.6.61	TCP	800	Forward	1335523149645
192.168.6.61	TCP	Hello	Forward	1335523149650
192.168.6.61	TCP	800	Forward	1335523149655
192.168.6.61	TCP	World	Forward	1335523149660
192.168.6.61	TCP	800	Forward	1335523149665
192.168.6.61	TCP	Hi	Forward	1335523149700
192.168.6.61	TCP	1000	Forward	1335523149705
192.168.6.61	TCP	Hello	Forward	1335523149710
192.168.6.61	TCP	1000	Forward	1335523149715
192.168.6.61	TCP	world	Forward	1335523149720
192.168.6.61	TCP	1000	Forward	1335523149725

Fig 3: Sender Table

Source IP	Protocol	Packet	Status	Sender Time(micro Seconds)
192.168.6.61	TCP	Hi	Forward	1335523149640
192.168.6.61	TCP	800	Forward	1335523149645
192.168.6.61	TCP	Hello	Forward	1335523149650
192.168.6.61	TCP	800	Forward	1335523149655
192.168.6.61	TCP	World	Forward	1335523149660
192.168.6.61	TCP	800	Forward	1335523149665
192.168.6.61	TCP	Hi	Discarded	1335523149700
192.168.6.61	TCP	1000	Discarded	1335523149705
192.168.6.61	TCP	Hello	Discarded	1335523149710
192.168.6.61	TCP	1000	Discarded	1335523149715
192.168.6.61	TCP	world	Discarded	1335523149720
192.168.6.61	TCP	1000	Discarded	1335523149725

Fig: 4 Receiver Table after DPU based HCF check

Host name	TTL(Microseconds)	*Priority*
System 1	3	1
System 2	300	1
System 2	900	2
System 2	1000	3
System 3	500	1
System 4	600	1
System 5	550	1
System 6	700	1
System 7	850	1
System 8	800	1
System 9	900	1

Fig 5.Updated HCF Table

The TTL field forwarded by the receiver is accepted by the verification module. This TTL is checked with the TTL obtained from the IP2HC table. When the value is same, the packet is considered as legitimate or else, it is discarded.

The updated IP2HC table is forwarded to all the system as they will be having an updated IP2HC table of the other hosts in the network.

5. Advantages

1. The Proposed System reduces the resending process.
2. In transmission time, the legitimate user can use the alternative path.
3.The Proposed System accepts the dynamic path during transmission.

6. conclusion

In this paper, we present a hop-count-based filtering scheme that detects and discards spoofed IP packets to conserve system resources. Our scheme inspects the hop-count of incoming packets to validate their legitimacy. Using only a moderate amount of storage, DPU based HCF constructs a Dynamic IP2HC mapping table via IP address aggregation and hop-count clustering. A pollution-proof mechanism initializes and updates entries in the mapping table. We have known that HCF can remove more than 90% of spoofed traffic. Moreover, even if an attacker is aware of HCF, he cannot easily circumvent HCF. Though we have described some advancements in this paper using Dynamic Path Update (DPU) based HCF concepts, it may also need some future enhancements.

Some of the future enhancements are:

1. The packet size can be reduced so as to decrease the network traffic.
2. Dynamic clustering algorithm can be altered so that the packet size will be minimized.

DPU based Hop Count filtering Algorithm

DPU Based Hop Count Filtering Algorithm:

Required: Updated IP2HC table

Assumption: Both Sender and Receiver knows the Updated IP2HC table.

Process

1.Sender sends the legitimate packet along with the T_e(experimental threshold) to the routers.

2.Routers forwards the packet to the receiver by dynamic path allocation.

3.Receiver receives the packet and extracts the T_e from the packet.

4.Checks the T_e with the T_a (Actual Threshold calculated from the TTL field in the packet).

5.If it matches, it is regarded as legitimate packet or else it is forwarded to dynamic Hop Count Check().

dynamic Hop Count Check()

6.In Dynamic hop count check, the IP address of the packet is extracted and mapped to IP2HC table. The corresponding Thresholds of all priorities are extracted from the table.

7.The T_e is compared with each prioritized thresholds $T_{ep}[p]$ (obtained from IP2HC table).

```
While(Tep[p]!='\0')
  { if(Tep==Te)
      {S(accepted);
       top=p;
      }
    else
      {p++;
      }
```

8.After checking with all possibilities if the threshold matches, status S and current priority P_i is obtained.

9. IP2HC table is updated with current P_i as the top priority and broadcasted all the nodes in the network.

REFERENCES

[1] [1] Ali, Mohammad Zulkernine, and Hossam Hassanein “Packet Filtering Based on Source Router Marking and Hop-Count”,, 32nd IEEE Conference on Local Computer Networks,2007.

[2] [2] Guang Jin, Yuan Li, Huizhan Zhang, Jiangbo Qian, “A Pi2HC Mechanism against DDoS Attacks” Third International Conference on Communications and Networking in China (CHINACOM), 2008.

[3] [3] Biswa Ranjan Swain and Bibhudatta Sahoo, “Mitigating DDoS attack and Saving Computational Time using a Probabilistic approach and HCF method ” IEEE International Advance Computing Conference (IACC 2009) Patiala, India, 2009.

[4] [4].Bharathi KrishnaKumar , P.Krishna Kumar , and R. Sukanesh, “Hop Count Based Packet Processing Approach to Counter DDoS Attacks”, International Conference on Recent Trends in Information,Telecommunication and Computing, 2010.

[5] [5] H. Farhat, “Protecting tcp services from denial of service attacks”, Proceedings of the 2006 SIGCOMM Workshop on large scale Attack Defense. New York, NY, USA: ACM Press, 2006.

[6] [6] P. Ferguson and D. Senie “Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing”, In RFC 2267, January 1998.

[7] [7] V. Paxson “An analysis of using reflectors for distributed denial-of-service attacks”. ACM Computer Communication Review, Vol.31 (3), July 2001.

[8] [8] Douligeris, C.Mitrokotsa , “DDoS Attacks and Defense Mechanisms Classification ",A, Proceedings of 3d IEEE International Symposium, 14-17 Dec 2003.

[9] [9] Spetch, Stephen M. Lee, Ruby B , “Distributed Denial of Service: Taxonomies of Attacks , Tools, and Countermeasures",, Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems, 2004 International Workshop on Security in Parallel and Distributed Systems, September 2004.

[10] [10] Jelena Mirkovic, Alefiya Hussain, Sonia Fahmy,, Peter Reiher, and Roshan K. Thomas “Accurately Measuring Denial of Service in Simulation and Testbed Experiments”, ,IEEE Transactions On Dependable And Secure Computing, VOL. 6, NO. 2, April-June, 2009.

[11] [11] Abraham Yaar, Adrian Perrig, Dawn Song “StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense”, IEEE Journal of Selected Areas in Communication, vol. 24, no. 10, Oct. 2006.

[12] [12]. Haining Wang , Cheng Jin, and Kang G. Shin ”Defense Against Spoofed IP Traffic Using Hop-Count Filtering”, IEEE Transactions on Networking,vol.15,No.1,2007.

[13] [13]. Chen, Hwang, and ku, “Collaborative Detection of DDoS attacks over multiple network domains”, ,IEEE Transactions on parallel and distributed systems, tpds-0228-0806,2006.

[14] [14] Y. Fan, H. Hassanein, and P. Martin “Proactive control of distributed denial of service attacks with source router preferential dropping,”, in Proceedings of 3rd ACS/IEEE International Conference on Computer Systems and Applications, 2005.

[15] [15] M. Ma “Tabu marking scheme to speedup ip traceback”, Journal of Computer Networks, vol. 50, no. 18, pp. 3536–3549, 2006.

[16] [16] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, “Practical network support for ip traceback,” ACM SIGCOMM Computer Communication Review, vol. 30, no.4, pp. 295–306, 2000.

[17] [17] D. X. Song and A. Perrig, “Advanced and authenticated marking schemes for ip traceback.” in IEEE INFOCOM, 2001, pp. 878–886.

[18] [18] A. C. Snoeren, “Hash-based ip traceback,” in SIGCOMM ’01: Proceedings of the 2001 Conference on Applications,Technologies, Architectures, and Protocols for Computer Communications. New York, NY, USA: ACM Press, 2001, pp. 3–14.

[19] [19] R. stone, “Centertrack: An ip overlay network for tracking dos floods,” in Proceedings of 9th USENIX Security Symposium, 2000, pp. 199–212.

[20] [20] M. P. Thomer M. Gil, “Multops: a data-structure for bandwidth attack detection,” in Proceedings of 10th USENIX security Symposium, 2001.

[21] [21] C. Jin, H. Wang and K. G. Shin, "Hop-count Filtering: An Effective Defense Against Spoofed Traffic," In Proceedings of the 10th ACM Conference on Computer and Communications Security, Washington, D.C.,USA, October, 2003.

[22] [22] Srinath,D. and Janet,J. “A Survey of Routing instability with IPSpoofing on the Internet”,Asian Journal of Information Technology,Vol. 9, No. 3, pp. 154-158, 2010.

[23] [23]Shui Yu, Wanlei Zhou, Robin Doss, and Weijia Jia, “Traceback of DDoS Attacks using EntropyVariations”, IEEE Transactions on Parallel and Distributed Systems,Vol. 22, No. 3, pp. 412-425, March 2011.