© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 06 Issue: 03 | July-September 2015

Network System Protection Against Internet Protocol Spoofing Based Distributed Denial Of Services Attacks During Socket Based Packet Transmission

[1]Sushil Kumar, [2]Ms. Poonam Rani, Assistant Professor ,
Department Of Computer Science And Application, Cdlu Sirsa

Abstract- In this paper, DPHCF-RTT technique has been implemented and analysed for variable number of hops. Goal is to improve limitations of Conventional HCF or Probabilistic HCF techniques by maximizing detection rate of illegitimate packets and reducing computation time. It is based on distributed probabilistic HCF using RTT. It has been used in an intermediate system. It has advantage for resolving problems of network bandwidth jam and host resources exhaustion. MATLAB 7 has been used for simulations. Mitigation of DDoS attacks have been done through DPHCF- RTT technique. It has been shown a maximum detection rate up to 99% of malicious packets. IP spoofing based DDoS attack that relies on multiple compromised hosts in network to attack victim. In IP spoofing, IP addresses can be forged easily, thus, makes it difficult to filter illegitimate packets from legitimate one out of aggregated traffic. A number of mitigation techniques have been proposed in literature by various researchers. Conventional Hop Count Filtering or probabilistic Hop Count Filtering based research work indicates problems related to higher computational time and low detection rate of illegitimate packets.

Keywords- Distributed Denial of Service (DDoS), Time to Live(TTL), Round Trip Time (RTT), Packet Filtering, Hop Count, Hop Count Filtering (HCF), Distributed Probabilistic HCF (DPHCF), Conventional HCF (CHCF), Probabilistic HCF(PHCF), Intermediate System

[I] INTRODUCTION

Distributed Denial of Service (DDoS) is a large-scale, coordinated attack on availability of services of a victim system or any network based resource that is launched indirectly by compromised hosts on Internet. In this attack, attacker fills networks bandwidth with large amount of request packets that consumes bandwidth and makes it difficult for legitimate user to access service. It can be performed at network level, operating system level, and application level. IP spoofing based DDoS attacks pose a big threat to availability of services on Internet. Without being authenticated on Internet, any packet can be sent to anyone. Packet filtering is both a tool and a technique which is a building block of network security. It is a means to impose control on types of traffic permitted to pass from one IP network to another. It examines header of packet and makes a determination of whether to pass or reject packet based upon contents of header. Packet filters operates at network layer and transport layer of TCP/IP protocol.

In this paper, section II presents Packet Filtering Techniques, section III presents DPHCF-RTT technique, section IV presents Results and Discussion, and lastly, section V presents Conclusions.

[II] PACKET FILTERING TECHNIQUES

In Hop Count Filtering (HCF), hop count is number of hops a packet traverses while moving from sender to receiver destination that can be used to assess authenticity of packet. IP TTL field prevent packets from looping forever. TTL is introduced to specify maximum lifetime of each packet in Internet which is an 8-bit field in IP header. In this, hop is counted from TTL field in IP header, which is set by sender and then is stored in a table. Each intermediate router decrements TTL value of an in-transit IP packet by one before forwarding it to next-hop. When a packet reaches its destination, final TTL value is initial TTL decreased by number of intermediate hop counts.

The packet is discarded when TTL reaches zero or when major difference occurs in number of hops in table in case of attack. Although any field in IP header can be forged by an attacker, he cannot falsify number of hops an IP packet takes to reach its destination. An attacker cannot randomly spoof IP addresses while maintaining consistent hop-counts as hop-count values are diverse. server can distinguish spoofed IP packets from legitimate ones using a mapping between IP address and hop counts. Source IP address spoofmg is a technique of lying about return address of a packet. With this, attackers can gain unauthorized access to a computer or a network by spoofing IP address of that machine.

The hop-count distribution of client IP addresses at a server takes a range of values for effective HCF. It is important to examine hop-count distributions at various locations in Internet as HCF cannot recognize forged packets whose source IP addresses have same hop-count value as that of an attacker. Those end systems would suffer only during an actual DDoS attack whose filter starts to discard packets only upon detection of a DDoS attack [5].

Ayman Mukaddam et al. [6] has proposed for victim side and conventional method of HCF has been used which is time consuming and not effective. Xia Wang et al. [7] are not trying to improve packet filtering technique which is needed for elimination of random IP spoofing. algorithm of Krishna Kumar et al. [1] requires a shared key between every pair of adjacent routers which requires lot of computational time and more than usual memory space. probability based hop count filtering (PHCF) technique ofB.R. Swain et al. [2] does not guarantee that remaining unchecked packets will be legitimate only. Hence, this technique lacks in maximizing up to 100% detection of illegitimate packets from total packets. In technique of Haining Wang et al. [5] attacker may also find effective way by creating an effective IP2HC table to overcome HCF. Hence, this is also ineffective as legitimacy of packets is not sure [8].

Hence, after reviewing literature, it is found that CHCF and PHCF techniques, which are used to filter malicious packets from total packets, possess certain limitations pertaining to computational time, detection rate of illegitimate packets. Hence, there exists lot of scope to maximize detection rate of illegitimate packets and reducing computational time.

[III] DPHCF-RTT TECHNIQUE

Usually, in conventional HCF 90% of malicious packets are dropped and in probabilistic HCF 80% to 85% of packets will be dropped but DPHCF-RTT, drops almost 100% of malicious packets. In DPHCF-RTT, focus has been kept on applying probability based distributed HCF along with RTT at each intermediate node and every packet has been checked once for its legitimacy at routers and then packet are transferred to victim side. details of this technique can be found in [9].

The malicious packets have been efficiently detected at intermediate routers through DPHCF-RTT technique. malicious packets, so discarded, do not contain any legitimate packets. number of packets, definitely, remains unchecked considering some threshold value of packets to be malicious in total number of packets. checked packets are passed to victim server and unchecked packets are passed on to next router for further application of DPHCF- RTT technique on them. This process is carried out till no unchecked packet remains.

The effectiveness of DPHCF-RTT has been examined over PHCF and CHCF technique in respect of detection rate of malicious or illegitimate packets and computation time for filtering malicious packets [9]. In this paper, DPHCF-RTT technique has been analysed extensively to observe impact of variable number of hops.

[IV]PROPOSED WORK

Socket Based Implementation

The endpoint in an interprocess communication is called a socket, or a network socket for disambiguation. Since most communication between computers is based on Internet Protocol, an almost equivalent term is *Internet socket*. data transmission between two sockets is organized by communications protocols, usually implemented in operating system of participating computers. Application programs write to and read from these sockets. Therefore, network programming is essential for socket programming.

Client server Model

It is possible for two network applications to begin

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 06 Issue: 03 | July-September 2015

simultaneously, but it is impractical to require it. Therefore, it makes sense to design communicating network applications to perform complementary network operations in sequence, rather than simultaneously. server executes first and waits to receive; client executes second and sends first network packet to server. After initial contact, either client or server is capable of sending and receiving data.

Overview of IP4 addresses

IP4 addresses are 32 bits long. They are expressed commonly in what is known as dotted decimal notation. Each of four bytes which makes up 32 address are expressed as an integer value (0 – 255) and separated by a dot. For example, 138.23.44.2 is an example of an IP4 address in dotted decimal notation. There are conversion functions which convert a 32 bit address into a dotted decimal string and vice versa. Often times though IP address is represented by a domain name, for example, hill.ucr.edu. Several functions described later will allow you to convert from one form to another (Magic provided by DNS!).The importance of IP addresses follows from fact that each host on Internet has a unique IP address. Thus, although Internet is made up of many networks of networks with many different types of architectures and transport mediums, it is IP address which provides a cohesive structure so that at least theoretically, (there are routing issues involved as well), any two hosts on Internet can communicate with each other.

Receiving DatagramPacket by DatagramSocket

```
//DReceiver.java

import java.net.*;

public class DReceiver{

  public static void main(String[] args) throws Ex
ception

{

   DatagramSocket ds = new DatagramSocket(30
00);

   byte[] buf = new byte[1024];

   DatagramPacket dp = new DatagramPacket(bu
f, 1024);

   ds.receive(dp);

   String str = new String(dp.getData(), 0, dp.get
Length());

   System.out.println(str);

 ds.close();   }}
```

Receiver checking ip details of sender

```
import java.net.*;
public class DReceiver{
 public static void main(String[] args) throws
Exception
{
  DatagramSocket ds = new DatagramSocket(3000);
  byte[] buf = new byte[1024];
  DatagramPacket dp = new DatagramPacket(buf,
1024);
  ds.receive(dp);
  String str = new String(dp.getData(), 0,
dp.getLength());
String info=dp.getAddress().toString();   // get ip
of sender
  System.out.println(str + " from " +info);
  ds.close();   }}
```

Receiver checking socket details of sender

```
import java.net.*;
public class DReceiver{
 public static void main(String[] args) throws
Exception
{
  DatagramSocket ds = new DatagramSocket(3000);
  byte[] buf = new byte[1024];
  DatagramPacket dp = new DatagramPacket(buf,
1024);
  ds.receive(dp);
  String str = new String(dp.getData(), 0,
dp.getLength());
String info=dp.getSocketAddress().toString();
  System.out.println(str + " from " +info);
  ds.close();   }}
```

[V] RESULT AND DISCUSSION

A. Detection Rute

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 06 Issue: 03 | July-September 2015

DPHCF-RTT technique has been examined on different hops from 1 to 4 as shown in Fig. 1. It is found more efficient when larger numbers of intermediate nodes are considered. It gives high detection rate of malicious packets as compared with lesser number of hops i.e. below 4. It has shown efficient results in getting detection rate of malicious packets up to 99.33%. Its detection rate consistently swings around optimum value of 99% which is a good sign of packet filtering. trend lines for variable number of hops shows that with increase in sample size of packets, detection rate of malicious packets has been increased. It is due to fact that packet flooding has created an impact of incorrect guessing of RTT values in combination with hop count values by attacker. With increase in utilization of number of hops and sample size of packets, rate of detection of malicious packets becomes consistent. It is due to fact that packet flooding ratio decreases with increase in utilization of number of hops for limited number of maximum packets size.

		Arrival rate of packets						
		1	2	3	4	5	6	7
	HOP1	25	35	37	40	57	80	85
Detection rates	HOP2	55	70	75	75	97	100	90
of Malicious packets(%)	HOP3	80	97	100	95	97	100	95
	HOP4	95	97	100	97	100	100	100

Table 1

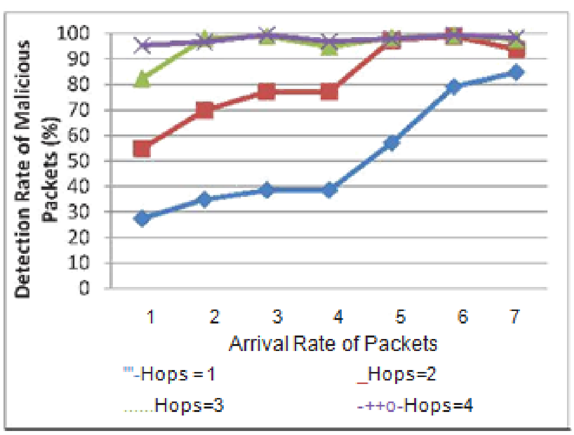

Fig. 1 DPHCF-RTT (Hops - 1 to 4)

Larger number of packets flooded larger will be effective increase in detection rate of malicious packets for different number of hops used. comparison of effectiveness and efficiency of DPHCF-RTT technique has been shown in Fig. 2 in between number of hops = 4 and 30.

		ARRIVAL RATE OF PACKETS						
		1	2	3	4	5	6	7
HOP	4	95	96	98	96	97	98	97
COUNT	30	100	100	99	98	99	100	97

Table 2

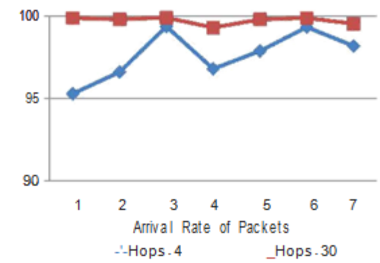

Fig. 2 DPHCF-RTT (Hops = 4 and 30)

In DPHCF-RTT, Packet Statistics (PS) has been determined under several hop conditions as mentioned in Table 1. Comparison has been done between proposed DPHCF-RTT technique at different hops, and existing PHCF technique at victim server.

This Packet Statistics is being described as follows:

- Total Malicious and Non-Malicious Packets used (M)
- Total Malicious Packets introduced (m) Probability based Total Malicious Packets (n) No. of Malicious Packets Detected (Count)
- Unidentified Malicious Packets being sent to Victim
- Server (m-Count)
- Malicious Packet Detection Rate (r)

DPHCF-RTT technique has shown a significant increase in its performance in detection rate of malicious packets when 30 numbers of intermediate hops, which is its maximum limit, have been considered over PHCF at victim server for total

400000 packets. Results of hops from 5 up to 29 have not been shown as there is no significant change occurred in them with respect to detection rate of malicious packets when compared with results for number of hops = 4. reason is limitation of number of packets considered up to 400000. If number of packets being flooded be increased, significant increase in detection rate of malicious packets can be obtained.

Computation Time

DPHCF-RTT has also been implemented for minimizing computation time for packet filtering. This technique filters malicious packets with minimum computation time as compared to PHCF and CHCF techniques at victim server for different number of hops at different arrival rate of packets. Hops = 1 in DPHCF-RTT takes very less computation time in contrast to more number of hops. This is due to high packet flooding and faster detection rate for a single hop. But, accuracy of detection is not sure in contrast to more number of hops. As number of hops increases, there is an obvious increase in total amount of time taken using DPHCF-RTT for lesser number of arrival rate of packets. But if arrival rate of packets is larger, then computation time for DPHCF-RTT is similar for different number of hops. In Fig. 3, comparison has been done between DPHCF-RTT for number of hops = 4 with PHCF and CHCF techniques at victim server.

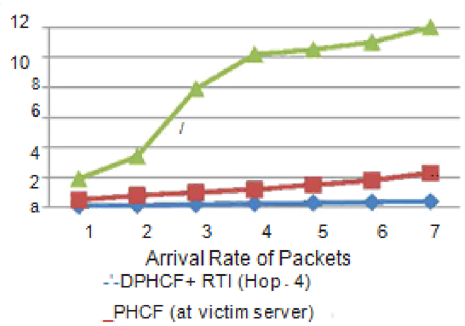

Fig **3**. DPHCF-RTT vs. PHCF and CHCF (Hops = 4)

	1	2	3	4	5	6	7
DPHCF+RTT	2	3	8	10	10.5	11	12
PHCF	1	1.2	1.4	1.5	1.6	1.7	2
CHCF	1	1	1	1	1	1.1	1.2

Table 3

[VI] CONCLUSION

Proposed DPHCF-RTT technique has been implemented. Its performance has been compared with PHCF and CHCF techniques. Detection rate of malicious packets and computation time have been considered as basis of comparison. Detection rate of malicious packets has been increased to 99% as compared to PHCF and CHCF techniques. Also, computation time for filtering illegitimate packets has been reduced drastically and has been proved effective as compared to PHCF technique. DPHCF-RTT can be implemented on real-time environment or on cloud platform for maximum number of intermediate nodes up to 30 in future.

Reference

1. Krishna Kumar, P.K. Kumar, R. Sukanesh, "Hop Count Based Packet Processing Approach to Counter DDoS Attacks," International Conference on Recent Trends in Information, Telecommunication and Computing, PET Engineering College, Thirunelvelli, India, pp. 271-273, 12-13, March,2010.
2. R. Swain, B. Saboo, "Mitigating DDoS attack and Saving Computational Time using a Probabilistic approach and HCF method," IEEE International Conference on Advance Computing,NIT, Rourkela, pp. 1170-1172,6-7, March 2009.
3. Mukaddam, I. H. Elhajj, "Hop count variability," 6th IEEE International Conference on Internet Technology and Secured Transactions, American University of Beirut, Lebanon,pp. 240-244, 11-14, December ,2011.
4. F. Zhang, J. eng, Z. Qin, M. Zhou, "Detecting DDoS Attacks Based on SYN proxy and Hop-Count Filter," IEEE International Conference on Communications, Circuits and Systems, University of Electronic Science and Technology, China,pp. 457-461,11-13,July,2007.

5. H. Wang, C.Jin and K. Shang, "Defense Against Spoofed IP Traffic Using Hop-Count Filtering," IEEE Transaction on Networking,vol. 15 (1),pp. 40-53,February,2007
6. Mukaddam, I. H. Elhajj, "Round Trip Time to Improve Hop Count Filtering," IEEE Symposium on Broadband Networks and Fast Internet, American University of Beirut, Lebanon,pp. 66-72,28-29, May,2012.

7. A Wang, Xia, Li Ming, Li Muhai, "A scheme of distributed hop-count filtering of traffic," International Communication Conference on Wireless Mobile and Computing, pp. 516-521, 7-9 Dec.2009.

8. Ritu Maheshwari, C. Rama Krishna, M. Sridhar Brahma "Distributed Denial of Service (DDoS) Attacks Mitigation and Packet Filtering Techniques: A Comprehensive Review," PTU National Conference on Innovations & Knowledge Discovery in Computing Technologies, lET Bhaddal, Punjab, India,pp. 9-15,13th_14th August,2013.

9. Ritu Maheshwari, C. Rama Krishna, "Mitigation of DDoS Attacks using Probability based Distributed Hop Count Filtering and Round Trip Time," International Journal of Engineering Research & Technology, vol. 2(7), pp. 1135-1140,July,2013.