DDoS Attack Prevention In Cloud Computing Using Hop Count Based Packet Monitoring Approach

Nisha H Bhandari[#]

[#]*Department of Computer Science & Technology, Gujarat Technological University*
Ahmedabad
[1]b_nish@yahoo.com

Abstract— **In cloud environment, cloud servers providing requested cloud services, sometimes may crash after receiving huge amount of request. This situation is called Denial Of service attack. Cloud computing is one of today's most exciting technologies due to its ability to reduce costs associated with computing while increasing flexibility and scalability for computer processes Recently Distributed Denial of Service (DDoS) attacks on clouds has become one of the serious threats to this buzzing technology. Distributed Denial-of-Service (DDoS) attacks are a significant problem because they are very hard to detect, there is no comprehensive solution and it can shut an organization off from the Internet. DoS attack is accompanied by IP Spoofing so as to hide the source of flooding and to make every request look different. The primary goal of an attack is to deny the victim's access to a particular resource.**

In this paper, we present an approach for packet monitoring in Cloud Environment to prevent DDoS attacks. This new approach of Hop Count Filtering provides a network independent and readily available solution to prevent DoS attack in Cloud environment. Also, this method decreases the unavailability of cloud services to legitimate clients, reduces number of updates and saves computation time.

Keywords— Cloud Computing, Distributed Denial of Service (DDoS) attack ,TTL, Hop-count ,packet marking.

I. INTRODUCTION

Cloud computing is currently one of the most hyped information technology fields and it has become one of the fastest growing segments of IT. Cloud computing allows us to scale our servers in magnitude and availability in order to provide services to a greater number of end users. Moreover, adopters of the cloud service model are charged based on a pay-per-use basis of the cloud's server and network resources, aka utility computing.Cloud computing is a model of information processing, storage, and delivery in which physical resources are provided to clients on demand. Instead of purchasing actual physical devices servers, storage, or any networking equipment, clients lease these resources from a cloud provider as an outsourced service. It can also be defined as “management of resources , applications and information as services over the cloud (internet) on demand”. Cloud computing is a model for enabling convenient and on demand network access to a shared group of computing resources that can be rapidly released with minimal management effort or service provider interaction. [1] Cloud computing provides different layers of computing utilities, from storage and networking to tools and applications, through three main service models: software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS).

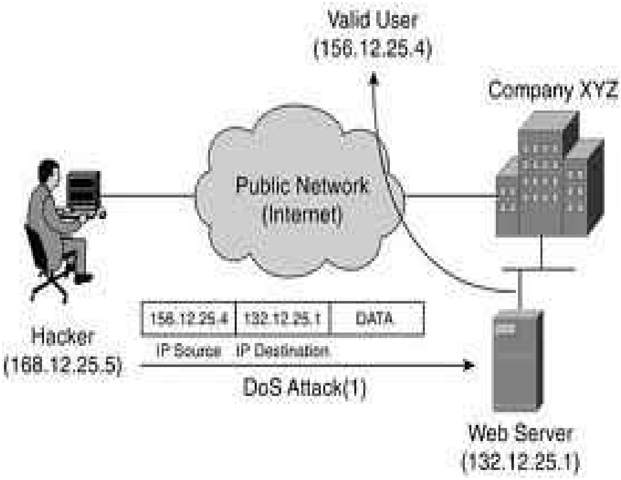

Figure 1 DoS Using IP Spoofing

DoS attacks do not wish to modify data or gain illegal access, but instead they target to crash the servers and whole networks, disrupting legitimate users’ communication. DoS attacks can be launched from either a single source or multiple sources. Multiple source DoS attacks are called distributed denial-of service (DDoS) attacks [2]. These attacks are a type of Flooding Attack [2, 3], which basically consist of an attacker sending a large number of nonsense requests to a certain service, which is providing various services under cloud. As each of these requests has to be handled by the service implementation in order to determine its invalidity, this causes a certain amount of workload per attack request, which in the case of a flood of requests usually would cause a Denial of Service to the server hardware [2].

II. HOP-COUNT FILTERING

Since hop-count information is not directly stored in the IP header, one has to compute it based on the Time-to-live (TTL) field. TTL is an 8-bit field in the IP header, originally introduced to specify the maximum lifetime of each packet in the Internet. Each intermediate router decrements the TTL value of an in-transit IP packet by one before forwarding it to the next-hop [4,5].

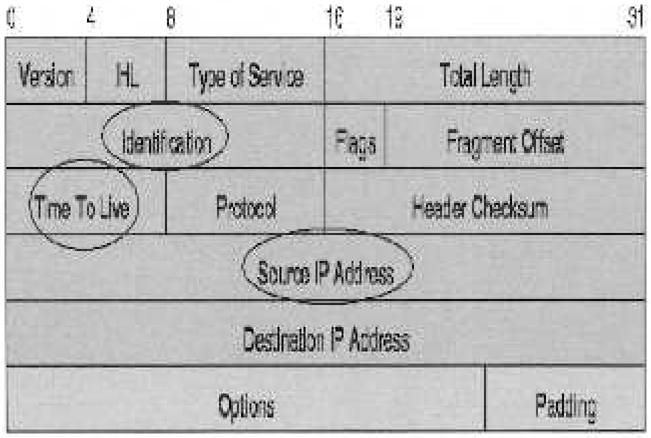

Figure 2 IP Packet Header

A. *Extract final value of TTL*

When a Packet reaches its destination and extracting its TTL field value, this value is known as final TTL. The challenge in hop-count computation is that a destination only sees the final TTL value. It would have been simple had all operating systems (OSs) used the same initial TTL value, but in practice, there is no consensus on the initial TTL value. Furthermore, since the OS for a given IP address may change with time, we cannot assume a single static initial TTL value for each IP address [4].

B. *Investigate the initial value of TTL*

According to [4], most modern OSs uses only a few selected initial TTL values, 30, 32, 60, 64, 128, and 255. Only a few Internet hosts are apart by more than 30 hops, thus one can determine the initial TTL value of a packet by selecting the smallest initial value in the set that is larger than its final TTL. For example, if the final TTL value is 112, the initial TTL value is 128, the smallest of the two possible initial values, 128 and 255. Thus, given the final TTL value one can find the initial TTL value. Initial TTL values can be calculated as follows [6]:

Initial TTL=32 if final TTL <=32
Initial TTL =64 if 32<final TTL<=64
Initial TTL =128 if 64<final TTL <=128
Initial TTL =255 if 128<final TTL <=255

C. *IP2HC Table*

The IP2HC table [6] is a mapping between Source IP Address of a packets and stored hop count for that IP Address. It is a structure with Source IP address serving as index to match the hop count information.

III. PROPOSED ALGORITHM

We The proposed algorithm, uses the hop count filtering mechanism, and provides a clear idea of implementation so that it can be used in Cloud environment to prevent DoS attacks. The algorithm requires continuous monitoring of packets travelling over the network in the Cloud, and thus, we extract SYN flag, TTL and source IP information from these monitored TCP/IP packets. The algorithm recognises four cases for each captured packet in the whole operation.

- If source IP address exist and SYN flag is set (Src=1 and SYN=1) in IP2HC table then calculate hop-count by using TTL value of IP packet. Now check if the hop-count matches with the stored hop-count, if not, then update source hop-count field of table for that source IP address.
- If source IP address exists and SYN flag is not set (Src=1 and SYN=0) in IP2HC table then calculate hop-count and if this hop count does not matches the stored hop count entry in the IP2HC table for the corresponding source IP address, then packet is spoofed, else the packet is legitimate.
- If source IP address does not exists and SYN flag is set (Src=0 and SYN=1) in the IP2HC table then calculate hop-count and add a new entry for the Source IP address with the corresponding hop count in the IP2HC table.
- If source IP address does not exists and SYN flag is not set (Src=0 and SYN=0) in IP2HC table then it means that the packet is spoofed, because every legitimate IP address having a valid TCP connection will have its entry in the IP2HC table.

ALGORITHM -1

Consider the following notations:
synflag = Syn bit of TCP packet.
mcount =malicious packet counter.
Tf= final value of TTL.
Ti=initial value of TTL.
Hs=stored Hopcount.
Hc=Computed Hopcount.(Hc=Ti-Tf)

```
START
   Initialize mcount=0;
    For each packet

   Set TTL = ExtractFinalValueOfTTL( );
   //get time-to-leave field of IP packet
    Set srcIp = ExtractSourceIP( );
   //get source IP address from IP packet
    Set synflag = ExtractSynBit( );
               //get Syn flag value from TCP packet

If  srcIp is exist in IP2HC table  then
    if  SYN flag is set
         Compute Hop Count (Hc) and compare with
         stored one (Hs);
         If  (Hc==Hs)
          allow the packet;
         else
         update hop-count value in IP2HC;

   else    //if SYN flag is not set
          Compute Hop Count (Hc) and compare
          with stored  one(Hs);
         If  (Hc==Hs)
         allow the packet;
         else
         drop the packet ;
         mcount++;

else  // srcIp  is not exist
    if  SYN flag is set
         Compute Hop count (Hc) ;
           Add entry into IP2HC table (srcIp,Hc);
    else //SYN flag is not set
           drop the packet;
          mcount++;
END
```

Here mcount gives total no of malicious packets. After completion of given algorithm spoofed packets are discarded and IP2HC table is updated successfully. It successfully reduces the no of updates in table and hence saves computation time by analyzing SYN flag of TCP protocol.

IV. CONCLUSION

Cloud Computing is gaining popularity, but with the widespread usage of cloud the issue of cloud security is also surfacing. One of the major threats to Cloud security is Distributed Denial of Service Attack (DDoS) or simply Denial of service attack (DoS). To improve resource availability of resources, it is essential to provide a mechanism to prevent DDoS attacks. One of the methods for prevention is Hop Count Filtering method (HCF). This paper presented a version of Hop Count filtering method which not only detects malicious packets but also includes update of IP to Hop count Table (IP2HC) with a mechanism that reduces the number of updates and thus saves computation time by analyzing SYN flag of TCP protocol.

References

[1] Ayesha Malik, Muhammad Mohsin Nazir Security Framework for Cloud Computing Environment: A Review Journal of Emerging Trends in Computing and Information Sciences VOL. 3, NO. 3, March 2012

[2] D. GARG, "DDOS Mitigation Techniques-A Survey," in International Conference on Advanced Computing, Communication and Networks, 2011.UACEE '11, pp. 1302-1309.

[3] S. T. a. K. Levitt, "Detecting spoofed packets," in Proceedings of The Third DARPA Information Survivability Conference and Exposition (DISCEX III) '2003, Washington, D.C., 2003.

[4] W. Haining, et al., "Defense Against Spoofed IP Traffic Using Hop-Count Filtering," Networking, IEEE/ACM Transactions on, vol. 15, pp. 40-53, 2007

[5] I. B. Mopari, et al., "Detection and defense against DDoS attack with IP spoofing," in Computing, Communication and Networking, 2008. ICCCn 2008. International Conference on, 2008, pp. 1-5.

[6] N. Venkatesu, et al., "An Effective Defense Against Distributed Denial of Service in GRID," in Emerging Trends in Engineering and Technology, 2008. ICETET '08. First International Conference on, 2008, pp. 373-378.

[7] P. S. Mann and D. Kumar, "A Reactive Defense Mechanism based on an Analytical Approach to Mitigate DDoS Attacks and Improve Network Performance," International Journal of Computer Applications, vol. 12-No.12, pp. 43-46, January 2011.

[8] S. T. a. K. Levitt, "Detecting spoofed packets," in Proceedings of The Third DARPA Information Survivability Conference and Exposition (DISCEX III) '2003, Washington, D.C., 2003.