© 2018 IJSRST | Volume 4 | Issue 9 | Print ISSN : 2395-6011 | Online ISSN : 2395-602X
Themed Section: Science and Technology

Analysis on DDoS Attack using a Probabilistic Approach

Dr. B. Mukunthan Ph.D[1], S. Dhayananthi M.C.A., B.Ed[2]

[1]Research Advisor, Jairam's Arts and Science College,Karur, Tamilnadu, India

[2]Research Scholar, Jairam's Arts and Science College,Karur, Tamilnadu, India

ABSTRACT

In the concepts of DDoS attackers, the researchers have taken two approaches. The first approach improves the routing infrastructures whereas the second approach protects Internet servers with sophisticated resource manage-ment to the servers. This end system approach provides more accurate resource accounting, and fine-grained service isolation and differentiation for example, to shield interactive video traffic from FTP traffic. However, without a mechanism to detect spoofed traffic, spoofed packets will share the same resource principals and code paths as legitimate requests. While a resource manager can confine the scope of damage to the particular service under attack, it cannot sustain the availability of that service. In stark contrast, the server's ability to filter most, if not all, spoofed IP packets can help sustain service availability even under DDoS attacks. Since filtering spoofed IP packets is orthogonal to resource management, it can also be used in conjunction with advanced resource-management schemes.

Keywords: DoS, Hop-Count Filtering, Hop-Count Inspection, spoofed, ns2, matlab

I. INTRODUCTION

A Denial of Service (DoS) attack can be characterized as an attack with the purpose of preventing legitimate users from using a victim computing system or network resource [1]. A Distributed Denial of Service (DDoS) attack uses many computers to launch a coordinated DoS attack against one or more targets. Using client/server technology, the perpetrator is able to multiply the effectiveness of the Denial of Service significantly by harnessing the resources of multiple unwitting accomplice computers which serve as attack platforms [2].

The rationale behind hop-count filtering is that most spoofed IP packets, when arriving at victims, do not carry hop-count values that are consistent with legitimate IP packets from the sources that have been spoofed. Hop-Count Filtering (HCF) builds an accurate HCI (IP to hop-count) mapping table, while using a moderate amount of storage, by clustering address prefixes based on hop-count. It is vitally important to prevent your systems from being used to launch these attacks. The compromised systems are usually taken over by one of two methods, namely password cracking and buffer overruns [3]. But very rare proposals have been made to solve this problem without using the support from network. In HCF (Hop-count filtering) method they have focused upon the value of TTL in [4] in which an IP packet header is checked to know the TTL value, whether the packet is malicious or not. A DDoS attack is composed of four elements in [5]:

- The real attacker.
- The handlers or master compromised hosts, who are capable of controlling multiple agents.
- The attack daemon agents or zombie hosts, who are responsible for generating a stream of packets toward the intended victim.
- A victim or target host.

These are the parameters that define the network QoS both within the network and at the edge access points where customer services are offered, leading to significant improvement in the network management [6]. If providers chose to charge differently for the use of different resources, they could charge for access to certain services within their networks. This would allow the providers to only allow legitimate customers on to their networks. This system would make it easier to prevent attackers from entering the network [7]. By altering the pricing of services, secondary victims who would be charged for accessing the Internet may become more conscious of the traffic they send into the network and hence may do a better job of policing themselves to verify that they are not participating in a DDoS attack.

Two running states, alert and action, within HCF use this mapping to inspect the IP header of each IP packet. Under normal condition, HCF resides in alert state, watching for abnormal TTL behaviors without discarding packets. Upon detection of an attack, HCF switches to action state, in which the HCF discards those IP packets with mismatching hop-counts. Besides the HCI inspection, several efficient mechanisms [8] are available to detect DDoS attacks.

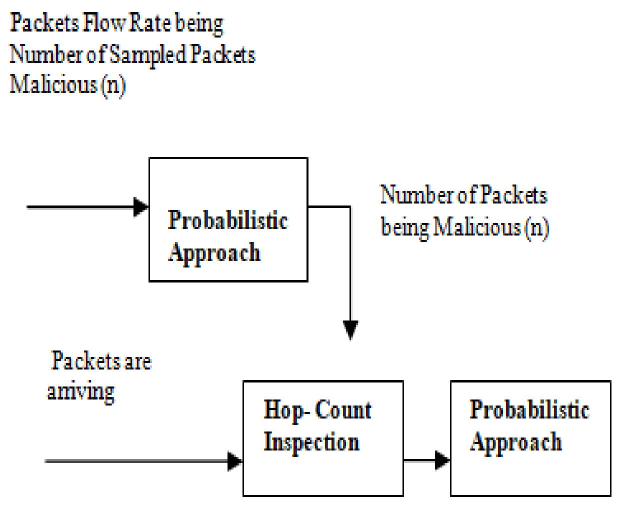
Packets Flow Rate being Number of Sampled Packets Malicious (n)
Probabilistic Approach
Number of Packets being Malicious (n)
Packets are arriving
Hop- Count Inspection
Probabilistic Approach

Figure 1. The Detailed design of Proposed System

II. PROPOSED SYSTEM

There already exist commercial solutions that block the propagation of DDoS traffic with router support. However, the main difference between our scheme and the existing approaches is that our approach & HCF are end-system-based mechanisms that do not require any network support. So we have compared the results of both the schemes. Our proposed approach saves the resource as the computational time needed by the system to mitigate in large extend, as the HCF method works in two phases.

In our approach we are calculating the number of malicious packets from a given number of packets using a probabilistic approach. We first sample the number of packets on the basis of arrival at the server per a time unit. By taking that number of packets, the average arrival rate of the packets and the error probability of a packet we calculate the number of packets being malicious. Then we apply the simple HCI (Hop-Count Inspection) Algorithm to filter out first that many number of packet how much was found out using the probabilistic approach. When we will reach at the exact number of packets, we simply release all the rest packets towards the server assuming that these are not malicious. It may also happen that we may lose some malicious packets being undetected but we save the computational time in a great extent than the actual HCF algorithm.

III. DETAILED STRUCTURE OVERVIEW

In our proposed view we are just showing the structure for only one server. this can be implemented on individual server in this manner. here in our structure we are having one victim server. in our structure we are having two important sections, i.e. the section which will implement the probabilistic approach and the section which will implement the Hop-Count Inspection (HCI) method. When packets are arrived, those will be sampled according to the willing of the victim. Then number of packet is taken for the sampling, their average arrival rate, and their probability of error in a packet will be taken into consideration for the calculation of number of packet being malicious by the probabilistic approach section. Then the probabilistic approach section will result out the value. Based on that value the Hop-Count Inspection section will filter out the packets which will be found malicious and dropped from being served by the server. Simultaneously a counter checks whether the number of found malicious packet has reached the number found by the probabilistic approach. If it will touch the limit very soon then all the packets remaining in the sampled packets will be released to the server without checking, whether those are malicious or not, assuming those are non-malicious in nature. By applying this approach we need not to check all the packets for the maliciousness. Hence we are saving the computational time with some extent and also mitigating the DDoS attack.

IV. PROPOSED ALGORITHM For Mitigation Of DDOS Attack

This algorithm is proposed to implement our approach at attacked end. Here initial input to the algorithm are the rate of arrival of the packets at the server, the error probability of each packet arrived at the server, and the number of packet was taken as a sample. Then these information were used to calculate the number of packets being malicious. That value is identified as 'n'. This value is input to the HCI section. In HCI section the packet will be checked whether the packet is 'spoofed' or not using the discussed TTL value of each packet. Depending on the result of HCI, the packet will either be allowed or dropped by the algorithm.

Table 1. Proposed Algorithm For Mitigation Of Ddos Attack

```
For given ' , ',' p','m+n':
Calculate 'n' such that P(N1=n,N2=m)=1;
integer count=0;
for each packet 'i':
if(count=6n)
status=HCI(i);
if status is 'spoofed'
count++;
drop the packet;
else
allow the packet;
end if;
end if;
end for;
```

The packet will be found *'spoofed'* then the counter value will be incremented and the packet will be dropped. The counter value is maintained because the counter will run up to *'n'* so that the first spoofed *'n'* packets will be dropped and rest of the sampled packet will be considered as non-malicious and allowed to the server. Each time a spoofed packet will be identified the *count* value will be incremented. Hence by using this approach we are saving the computational time which was not saved in classical HCF mechanism. In classical HCF method each and every packet were checked for the maliciousness and using two sections of execution and using threshold value.

V. SIMULATION RESULTS

Some portion of the analysis of this approach was simulated using ns2 and some portion using matlab. We first analyze the approach in ns2 by taking some nodes. Some of them were attackers and it was intending to attack on one victim. So we forward packets from those malicious nodes by spoofing the source address. And others were pretending them as non-malicious node and they were sending packets using their own source address. By applying the probabilistic approach and the HCI method at the victim side we caught some packets as malicious and some as non-malicious. So from there we have collected 16 set of data and maintained into an array. We have collected the dataset about the HCF method and use them for the analysis of the performance. Then we have just compared the results of our approach and the results of classical HCF method. Our approach took less time than the HCF method.

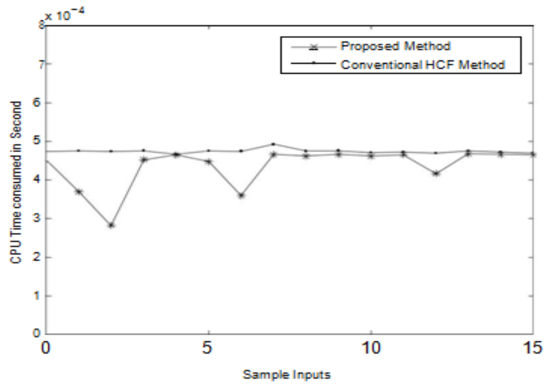

Figure 2. Comparison between time taken by Proposed Algorithm and HCF method (Packet flow rate=8000 packets/second)

We have collected the dataset of different flow rate of packets. And in our simulation we have used that flow rate to validate our approach. For our simulation we have taken the flow rate of packet as 8000 packets/second. From the above we can see the resource as time is saved by our approach. It takes less time than the HCF method because of the complexity of the execution of HCF method, which takes two steps in execution (*alert* and *action* states). Here we compared the sixteen data we got from our result.

In [4] it was given that the 90% of DDoS traffic was caught using the HCF method. Here also averagely 85% to 90% of DDoS traffic was caught. And that is because of that we are not using any queue for keeping the packets when they are reaching at the server side. We are using all those packets as they are coming to the server. That's why sometime the packet detection is less than 100% then it will come to 100% and again it will go less than 100%. This cycle will continue. This Simulation Results is the future enhancement to our proposed model. Very less amount of packets got undetected using this approach as we are using the probabilistic approach to solve that whereas the HCF method used the threshold for considering the packets to be malicious.

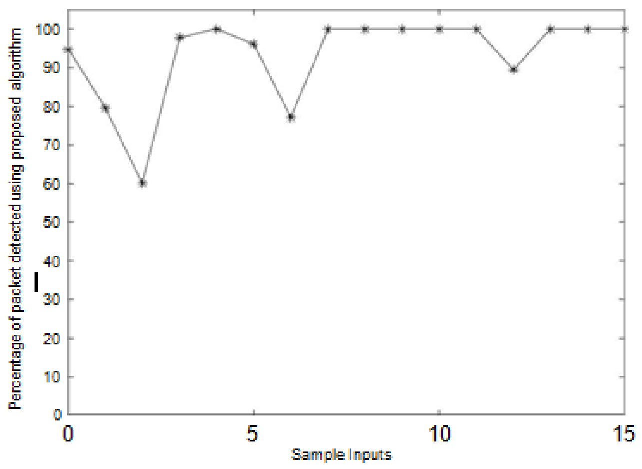

Figure 3. Percentage of packets detected using Proposed Algorithm (Packet flow rate=8000 packets/second)

From the following dataset we have compared our result with the existing one and simulated the same to show our proposed approach's performance. Figure 2 shows the comparison between the time taken by the HCF method and the proposed method. Figure 3 shows the packets being detected by the proposed method, and it shows averagely our approach shows about to 90% of DDoS packets are being detected.

We have collected so many dataset of HCF Method. Some of them are as follows,

Table 2. Packet Flow Rate (¸) =3000 Packets/Second

Sample	HCF Method (£ 10^{i3} sec)
1	0.1783
2	0.1767
3	0.2724
4	0.1761
5	0.1763

6	0.1764
7	0.1782
8	0.1832
9	0.1748
10	0.1770
11	0.1752
12	0.1758
13	0.4845
14	0.2753
15	0.2619
16	0.2623

Table 3. Packet Flow Rate (¸) =6000 Packets/Second

Sample	**HCF Method (£ 10^{i3} sec)**
1	0.3539
2	0.4458
3	0.3562
4	0.3523
5	0.3581
6	0.3529
7	0.3559
8	0.4187
9	0.3482
10	0.7400
11	0.5709
12	0.5440
13	0.5433
14	0.4057
15	0.3519
16	0.3581

Table 4. Packet Flowrate(¸)=10,000 Packets/Second

Sample	**HCF Method (£ 10^{i3} sec)**
1	0.5946
2	0.9060
3	0.9490
4	0.9608
5	0.9232
6	0.9656
7	0.9735
8	0.9619
9	0.9364
10	0.8407
11	0.5996
12	0.5822
13	0.7272
14	0.5948
15	0.5839
16	0.5840

VI. CONCLUSION

We got the idea that how much harmful this attack is and how much uncertain this attack is. As this attack is very much uncertain we proposed one probabilistic approach to work this out. Then that probabilistic approach shows us the way to solve the problem. Then simple HCI (Hop-Count Inspection) method made it possible to solve it out. As here we have compared the result of our approach with the result of HCF method, we have collected the dataset required to solve the problem. The simulation results shows that our approach is taking less amount of time than the HCF method and also it could detect about to 90% of the DDoS packets from all the packets.

VII. REFERENCES

[1]. S. Specht and R. Lee. "Taxonomies of distributed denial of service networks, attacks, tools, and countermeasures", Technical Report CE-L2003-03, 164, May 2003.

[2]. http://www.w3.org/security/faq/wwwsf6.html.

[3]. Graham Wheeler. Denial-of-service:courting disaster. Technical Re-port,CEQURUX Technologies.

[4]. Haining Wang , Cheng Jin and Kang G. Shin. Hop-count filtering: "An effective defense against spoofed traffic", Technical Report,The Pennsylvania State University, 2003.

[5]. Christos Douligeris and Aikaterini Mitrokotsa. Ddos attacks and defense mechanisms: "A classi¯cations. Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technolog"y, pages 190-193, December 2003.

[6]. http://www.wired.com/news/business/0,1367,34221, 00.html Yahoo on Trail of Site Hackers, February 2000.

[7]. Ceilyn Boyd John Zao David Mankins, Rajesh Krishnan and Michael Frentz. Mitigating distributed denial of service attacks with dynamic resource pricing. Proceedings of 17th Annual Conference on Computer Security Applications , 2001. ACSAC 2001, pages 411-421, 2001.

[8]. M. Fullmer and S. Romig. The osu °ow-tools package and cisco netflowlogs. In Proceedings of USENIX LISA'2000, New Orleans, LA, December 2000