Detection of DDoS Attack using Optimized Hop Count Filtering Technique

G. Usha Devi*, M. K. Priyan, E. Vishnu Balan, C. Gokul Nath and M. Chandrasekhar

School of Information Technology and Engineering, VIT University, Vellore – 632014, Tamilnadu, India; ushadevi.g@vit.ac.in, priyanit085@gmail.com, vishnubalan91@gmail.com, gokulkapoor@gmail.com, chandrasekharmogali@gmail.com

Abstract

Background: The Distributed Denial of Service (DDoS) attack is one of the most recent and most vulnerable attacks that can happen to a web server to crash or stop providing survive. Many papers have been proposed to resolve this attack and have resolved to a certain extent but it is very difficult to explore and solve every loophole since Internet is a vast domain.

Methods: In Denial of Services (DoS) attack, the attacker uses up all the resources available to the server so that the legitimate user does not get the actual service. The well-established network infrastructure is the backbone to carry out this attack. DoS attacks are very severe when it happens to important servers such as banking and government websites. In this paper, we have proposed a new optimized mechanism which could be more reliable than the existing models. The traffic generated by an IP packet is recorded and a window matrix is generated. This matrix consists of number of packets from each IP during a given window and the maximum packets received from an IP will be decided. Then, this result is used as an input to the Hop Count Filtering (HCF) algorithm, the packets can be distinguished as legitimate and attacker packets.

Results: This paper proposes a technique to detect Distributed Denial of Service attack by using window matrix and optimized HCF filtering technique. Finally, the algorithm says that the packets are legitimate IP packets and spoofed IP packets along with their IP addresses. Based on this result, we conclude to accept or discard the packets.

Applications: This paper combines the features of existing methods to track the IP address as in tracking applications and it is also used to discard the unwanted packets based on IP address.

Keywords: DDoS, Hop Count Filtering, IP spoofing, Optimized HCF, Window Matrix

1. Introduction

Internet technology has brought great transformation in our daily life. The positive, negative and worst have come out from its usage in the last two decades. Internet is being used as a tool to ease and expand the field of education, research, governance, business, entertainment. These are some of the widely seen positives. The monetary loss, mental or physical harassment, personal information being illegally collected and sold with various personal interests, defamation are some of the negative effects of it. The worst are those acts of humans for which there is no law up to date to punish them. The government and the internet security agencies need to work together to form the laws and punish the culprit. The acts can be broadly classified as technology centric and non-technology centric*. Phishing, Click fraud, Spam, Malware, Hacking and Denial of Service are some important technology centric attacks. In phishing the internet user is directed to a pseudo website of an original one where he is asked to enter his personal sensitive information. This information intern is used or sold later for personal benefits. In click fraud the number of clicks on online ads of the advertiser is falsified to cause monetary loss. Spam is the unwanted messages or notices. Most of them being the marketing or e-fraud. Some of them intended to draw to innocent people into trap and cause damages in different forms. Malware are computer programs which enter ones system from different paths. Once they enter the system, they start effecting the normal functioning of system resources or the network or some
even send the data (personal, financial related or password of social sites) from host to the attacker. Unauthorized access to ones system without the knowledge of authorized user happens in hacking. In Denial of Service Bogus requests are flooded to the victim machine so that it does not perform the actual intended work.

2. Distributed Denial of Service Attack

The attacker uses up all the resources available to the server so that the legitimate user does not get the actual service in Denial of Services (DoS) attack. The well-established network infrastructure is the backbone to carry out this attack. DoS attacks are very severe when it happens to important servers such as banking and government websites.

The Figure 1 shows the architecture of direct attack and reflects attack respectively. In the earlier DoS attacks, the attacker would use a single victim system to flood networks. But the attacks (Distributed Denial of Services) happen from very large number of systems present at different physical locations, so it is difficult to trace all the systems. Distributed Denial of Services (DDoS) attacks can be divided broadly into two forms on as application bug level and infrastructure level. In application bug level, a bug that uses loopholes in computing systems or a nicely crafted packet that overloads the application and hence leading them to crash. "Ping-of-death" is one of the classical examples. In infrastructure level, the robust network infrastructure is taken as flat form to take control of Zombie machines by the attacker. From these machines the packets are flooded to the victim. This is taking place in a distributed manner (Distributed Denial of Service). The attacker can set his attack in two ways, either direct or reflect attack.

2.1 The Direct Attack

The packets are flooded to the victim using Zombie machines. No intermediate machine is being used. It can be further classified as:

- Network-layer: The attacks occurring at network layer like SYN flood, TCP flood, UDP flood and ICMP flood.
- Application-layer: The attacks which are occurring at application layer like HTTP flood, FTP flood and HTTPS attack.

2.2 Reflect Attack

In reflect attack, the requests are being sent to reflector by the Zombie machines spoofing the source IP address of the victim server. The packets are amplified before flooding the victim here. Many papers have been published that gives different strategies used to detect DDoS attacks depending on basic parameters such as CPU usage, memory usage of the server, bandwidth of network during the attack period and normal period.

The DDoS attack is detected either at victim end or in the routers. Routers used to detect can be at source-end, core-end or at victim end in the network infrastructure. The routers close to the source of traffic are source end routers. The routers used as the high performance back bone routers in internet are core routers and those at victim are victim-end routers. DDoS defense techniques based on defense points can be established at source, core and victim-end or can be distributed. When defense technique is being employed at source router called source end defense, at core routers called core end defense and at victim end routers called victim end defense technique.
When the defense technique is distributed i.e., distribute employed at core, server and core end then it is distributed technique.

Defense techniques can also be employed according to the reaction time\(^1\). They are proactive, reactive and survival. The proactive technique is applied well in advance or may be as built in the routers to prevent the attack. The reactive technique fight the attack after the DDoS has occurred. In survival technique, the system has to survive the attack so that it increases its bottleneck resources. If the victim detects itself being in attack at its early stage then it gets more time to react efficiently and hence reduce the occurrence of damage.

### 3. Detailed Problem Definition
Currently, DDoS Attacks with spoofed IP Packets are increasing at an alarming rate. IP Spoofing is used to generate huge data traffic which is directed to a target machine such that it won't be able to serve the legitimate users and finally goes down. More attackers have gone for a new type of DDoS attack known as Distributed Reflection Denial of Service (DRDoS) which is feasible only with the help of IP Spoofing. In DRDoS, victim IP address is spoofed in the DNS Request Queries (1 Byte) and sent to the DNS Server which replies (8 Bytes) to the Victim machine which multiplies the traffic into manifolds thereby inflicting DDOS attack on the server. To Counterattack DDoS, research people have come up with lots of efficient and reliable defensive techniques. These defensive techniques can be applied at router level\(^1\) that consists of routers in the core network, victim end\(^2-4\), the targeted host machine and source end\(^5\), the machines which generate huge network traffic directed towards the target machine. Out of these, applying detection technique at the victim end comes with lesser complications as there is only one victim but whereas routers and sources can be many. At present, most of the victim end detection techniques go for Survival techniques wherein they increase their resource pool of server to handle incoming traffic. But this technique cannot withstand high inbound traffic as the resource extension possible up to a limit and it may exhaust the CPU resources very quickly thereby leaving less space for detection and recovery.

In this paper, we optimize the Hop count filtering technique by identifying the source IP of packets which form the major portion of attack traffic. The basic idea is to apply the HCF Mechanism, on packets which form a huge part of incoming attack data traffic. This technique doesn't guarantee 100% detection but it can drop most of the spoofed IP packets which form the attack traffic thus protecting the server from depletion of the resources. HCF Mechanism takes advantage of the IP header information which cannot be forged easily to differentiate between the spoofed and legitimate packets. We optimize the HCF mechanism by selectively choosing the packets from the attack traffic there by reducing the computational overhead of the HCF Mechanism. We selectively choose the Anti-DDoS (ANTID) and Trace back techniques are some examples. In this paper, DDoS attack detection with optimized hop count filtering technique is being proposed.

### 2.3 Proactive Techniques
Routers play a major role in this technique as victim server does not know itself that it was under attack. The routers detect the spiteful traffic automatically and filter out the malign packets. Some of the widely used techniques in this are Ingress/egress filtering, Route-based Distributed Packet Filtering (DPF), D-WARD, Proactive cooperative defense, Internet Indirection Infrastructure (i3), Secure Overlay Services (SOS) and it can collaborative detection of DDoS attacks.

### 2.4 Survival Techniques
This is the most broadly used technique at present. In this technique, the victim increases his resources like CPU power, memory, bandwidth and TCP buffer. They may be added statistically (by purchasing) or dynamically (redirecting to public servers). Its feasibility is too low as the attacker can go on increasing the set of zombie machines and more number of internet users added every day. Adding multiple proxy servers before the main server adds up large TCP buffer and hence it can handle SYN flood to certain extent. Backlog queue is enlarged by increasing the memory used to hold the SYN flags set until the way of handshake procedures are completed. By changing the timeout for connection requests and hence the buffer is released faster than the normal time without waiting for ACK packet.

### 2.5 Reactive Techniques
In this technique, the victim server is sole responsible to detect and fight against the attack. Most of the existing DDoS defense techniques belong to this category. Push Back, K-MaxMin, Hop Count Packet Filtering (HCF),
packets from the Window Matrix with the help of Genetic Algorithm (GA). HCF mechanism uses the fact that the hop count information in the IP header cannot be forged easily.

Hop count information can be inferred from the 8-bit Time-to-Live (TTL) field of header. As each and every hop between the source and the destination decrements the TTL value by one before forwarding to the next hop. The background behind the HCF Mechanism is that the spoofed IP addresses which arrive at the receiver will have hop count values which are not consistent with the spoofed IP addresses. There by facilitating in weeding out the forged packets and retaining the legitimate ones. But the HCF mechanism suffers from the drawback that it has to perform the procedure of calculating Hop Count Information for each and every packet thereby increasing the computational overhead. We take into the fact that when a large amount of traffic is received as part of the attack, a stream of packets carries the same IP address and then the HCF mechanism is applied for this particular IP address. We identify the particular IP address with the help of Window Matrix and Genetic Algorithm (GA). Like all other existing techniques even though we optimize the HCF Mechanism, it has its own limitations. If the attacker doesn't use the spoofed IP traffic, he cannot be trapped by our technique. But nevertheless, this optimized technique avoids a large chunk of attacker traffic which is quite sufficient is small end systems.

4. Proposed Detection Model

The Genetic Algorithm is used to optimize and determine a value for the input window size \((n)\) from the incoming data traffic. Generally the window size will be less when the incoming data traffic is high and vice versa. Then we go on to calculate the Window Matrix \((n \times n)\) for each window. The packets that are present in the window are mapped into the Window Matrix with help of a hash function which uses IP address as the input and then maps the particular packet onto the matrix. If “\(i\)” is the row of the matrix and “\(j\)” is the column then \(M(i, j)\) represents the value for a particular packet in the current window. If there are no collisions then the value is incremented for that particular packet. In the current Window Matrix, each non-zero value represents the number of packets from a particular source IP address. The hash function gives us co-ordinates required to map the IP address into the matrix. As the range of IPV4 address is very large, we cannot accommodate all the addresses in the matrix hence we reduce the matrix size with the help of the hash function. The collision in the hash function can be of little impact as the packets are mapped to a single matrix. Construction of Window Matrix uses hash function.

IPV4 address consists of 4 octets. We calculate the row index by using the lower 16 bits of the IP Address using the modular operation. The lower 16 bits of the IP is divided by the row size and the remainder is used as the index for the row. The higher 16 bits of the IP is divided by the column size and used as index for the column. The value corresponding to the index \((i, j)\) is incremented. If there are no collisions then the value indicates the number of packets from a specific IP address in that particular window.

Hop Count cannot be directly fetched from the IP header information. Time-to-Live (TTL) is an 8-bit field in the header and it is used to indicate the number of hops that the packet can travel. The value in the TTL field is decremented by every hop and it is forwarded to the next hop. But the problem in computing the hop count information is that the receiver sees only the final TTL value. All the Operating system used a constant Initial TTL value then it would have been easy to calculate the hop count information.

According to\(^6\), most of the operating systems use selected set of TTL values 30, 32, 60, 64, 128 and 255. We observe that all the values are far apart except 30, 32 and 60, 64. According to\(^7,8\), it is shown that some of the hosts are separated by more than 30 intermediate hops. So we determine the initial TTL value by choosing the smallest value from the above set which is greater than the final TTL value of the incoming packet. For example if the final TTL value of a packet is 8 then we can infer the initial TTL value as 30 but there is an ambiguity with the values like (30, 32), (60, 64) and (32, 60). In these types of scenarios, we compute the Hop count information values for both the initial TTL values and try to match the result with either of them. The drawback of using these set of initial TTL values is that, if any Operating system is using an odd TTL value then we determine it as a spoofed packet, even though it was legitimate. But according to defending against spoofed DDoS, these machines constitute a very less percentage in the current internet network.

There will be a mapping table maintained for the IP addresses and their corresponding Hop count
information. It is taken into the fact that the machine stores the IP Address–Hop count information of the hosts in the internet as most the hop count values are stable and any dynamic changes in the network topology is recorded by the machine with the help of trace route mechanism and keeps the mapping table updated and consistent with the current router and network topology. Now we will see how the packets are identified as spoofed and legitimate packets. After the construction of the Window Matrix, we find out the highest value in the matrix, which indicates highest number of packets have been received by that particular IP Address in that Window. Now we extract the IP Address and TTL value of that packet with largest value in the Matrix and give it as an input to the HCF Mechanism, which determines whether the packet is spoofed or legitimate. We guess the Initial TTL value (T1) for that packet from the above mentioned set which is the smallest value greater than the final TTL value (T2). Then we compute the difference between Initial TTL and Final TTL which gives the Hop count. Using the Source IP Address which serves as the Index to the mapping table and the Exact Hop count is retrieved. Then the calculated Hop count is matched with the Exact Hop count, if they match then the packet is legitimate otherwise it is spoofed. The Figure 2 shows the flow diagram of the proposed work.

4.1 The Algorithm
Step 1: For each window of size n
Step 2: Construct a window matrix of size n * n
Step 3: Get the IP address (IPmax) corresponding to the max value in the window matrix
Step 4: Extract the Final TTL (FT) from the source IP address IPmax
Step 5: Infer the initial TTL (IT)
Step 6: Compute hop count CH = IT - FT
Step 7: Index IPmax to get Stored Hop count (SH)
Step 8: If (CH! = SH), the packet and IP are spoofed
Step 9: Else the packet and IP are legitimate

5. Conclusion
In this paper, we propose an optimized hop count filtering technique which is an efficient algorithm to detect spoofed IP address and packets. We have suggested few changes in the existing algorithm which makes the current algorithm more reliable and optimized. The traffic matrix we used in this paper helps in tracking the IP address that has sent maximum number of packets to a server. Using this IP address, it can be concluded and decided as which packets are to be discarded. The proposed paper combines the advantages of both the approaches hence leading a new optimized approach.

6. References
1. Kim W, RanJeong O, Kim C, So J. The dark side of the Internet: Attacks, costs and responses. Information Systems. 2011; 36(3):675–705.
2. Sawwashere S, Srivastava S, Lanjewar A, Bhilare DS. Optimizing DDOS attacks using LCIA. International Journal of Application or Innovation in Engineering and Management (IJAEM). 2013; 2(12):11–17.
3. Preetha G, Kiruthika Devi BS, Shalimie SM. Autonomous agent for DDoS attack detection and defence in an experimental testbed. International Journal of Fuzzy Systems. 2014; 16(4):520–8.
4. Beitollahi H, Deconinck G. Analyzing well-known countermeasures against distributed denial of service attacks. Computer Communications. 2012; 35(11):1312–32.
5. Zhang G, Parashar M. Cooperative defense against DDoS attacks. Journal of Research and Practice in Information Technology. 2006; 38(1):69–84.
6. Lee F, Shieh S. Defending against spoofed DDoS attacks with path fingerprint. Computers and Security. 2005; 24:571–86.
7. Wang H, Jin C, Shin KG. Defense against spoofed IP traffic using hop-count filtering. IEEE/ACM Transactions on Networking. 15(1):40–53.
8. Leea SM, Kimb DS, Leea JK, Park JS. Detection of DDoS attacks using optimized traffic matrix. Computers, Mathematics and Applications. 2012; 63(2):501–10.
9. Mirkovic J, Reiher P. D-WARD: A source-end defense against flooding denial-of-service attacks. IEEE Transactions on Dependable and Secure Computing. 2005; 2(3):216–332.
10. The Swiss Education and Research Network, Default TTL values in TCP/IP 2002. Available from: http://secfr.nerim.net/docs/fingerprint/en/ttl_default.html
11. Cheswick B, Burch H, Branigan S. Mapping and visualizing the internet. Proceedings of USENIX Annual Technical Conference; 2000. p. 1–12.
12. Claffy K, Monk TE, McRobb D. Internet tomography. Nature. 1999 Jan 7.