International Journal of Computing Academic Research (IJCAR)
ISSN 2305-9184 Volume 3, Number 3(June 2014), pp. 77-87

http://www.meacse.org/ijcar

An Efficient Source Information based Filtering Scheme for DDOS Attacks

M.Parameswari and S.Sukumaran

Department of Computer Science, Erode Arts & Science College, Erode – 638 009, Tamilnadu, India

Abstract: These days, Internet is the most essential medium for communication which is used by many users across the Network. Together, its commercial nature is causing enhance vulnerability to increase cyber crimes and there has been an immeasurable raise in the number of Distributed Denial of Service (DDOS) attacks on the internet over the past decade. Resources of network such as web servers, network bandwidth and network switches are generally the victims of DDoS attacks. DDoS attack tools employed a lot of IP address spoofing. The majority of the recent research on DDoS attack packet filtering depends on cooperation among routers, which is tough to attain in real campaigns. Hence, in this paper to defend against various source IP address spoofing a novel filtering scheme is proposed based on source information. The proposed scheme works autonomously at the potential victim side, and gathers the source information of its clients, for instance, source IP addresses, skips from the server during attacks free period. When a DDoS attack alarm is raised, the attack packets can be filtered based on the gathered d knowledge of the legitimate clients. The source IP addresses is divided into $n (1 \leq n \leq 32)$ segments in the proposed algorithm; based on result, the challenge storage can be released and speed up the process of information retrieval. The proposed system and the experiments show that the method works effectively and efficiently.

Keywords: DDoS attack, Network Security, Intrusion Detection.

Introduction

Secured communication has various desirable security aspects like authentication, confidentiality, non repudiation and message integrity. Moreover, in recent times many people are aware that availability and access control are also urgent requirements of protected communication because of the disreputable Denial of Service (DoS) attacks that render by the unlawful users into a host, network, or other piece of network infrastructure to harm them, particularly it is done against the commonly visited websites of a number of prestigious companies or government websites. Distributed Denial of Service (DDoS) attack exploits adequate puppet computers to generate number of data packets, the attacks become coordinated and come from various puppets at a time thus are even overwhelming.

Internet design concentrates normally on providing functionality nevertheless a small concentration has been given on strategies designing for controlling unintentional failures. Conversely, planned attacks by malicious users/ crackers/ hackers have no respond in the original Internet design. A DoS (Denial of Service) is such a planned attempt by malicious attackers / users to completely degrade or disrupt availability of resource/service to authorized /genuine users [3]. Some familiar DoS attacks are teardrop, SYN Flood, smurf, land, finger bom, ping of death, octopus,black holes, snork, misdirection and the ARP Cache poisoning. DoS attacks utilize weaknesses in applications, Internet protocols and protocol accomplishment in operating systems. DDoS (Distributed Denial of Service) attacks corrupt or completely disturb services to authentic users by disbursing communication and/or resources of the target. Mirkovic et al. [2] illustrated DDoS attacks as improved form of DoS attacks, where attackers direct thousands of compromised hosts labeled zombies against a single target. These zombie hosts are harmlessly hired from

the millions of vulnerable computers accessing the Internet through always available connections and high-bandwidth.

A usual DDoS attack contains two phases, the first phase is to compromise vulnerable systems that are easily accessible in the Internet and attack tools are installed in these systems. It is known as turning the systems into "zombies." In the second phase, an *attack command* is send *to the "zombies"* by the attacker through a protected channel to initiate a bandwidth attack against the targeted victim(s). The security deficiency and vulnerability of the TCP/IP suite brings the beginning DDoS attacks on easily, so far it is tremendously hard to secure against them. Normally, there are two types in this field, attack packets filtering and invasion detection.

Generally the common IP spoofing types can be put in order into three as follows subnet spoofing, fixed spoofing and random spoofing. It is very difficult to make out attack packets from valid the Internet traffic, and further filter them when a DDoS attack happened. The route-based packet filters proposed by Lee and Park [3] as a type of justifying IP spoofing, which believes that there is one single path in between a source node and destination node, thus any packet with the source and the destination addresses that emerge in a router that is not in the specified path, should be discarded. This problem is also tried to solve by Packet marking [2], [4] method. The fundamental idea is that routers on the attack graph mark the packets those are passing via the local router, a fingerprint will be created for an attack packet by the assistance between the routers those are placed on the attack path. The victim observes the fingerprints by the source IP address, and then recognizes the spoofed packets. These techniques have a number of necessary flaws which restrict their application in the actual clash against DDoS attacks. Initially, it requires all the routers' cooperation on the path of attack, which is obviously difficult to fulfill on the Internet. Besides, victims should understand the topology of whole network, which is not possible for huge volume of DDoS attacks. Thus, in order to improve efficiency and feasibility against DDoS of spoofed address, an advanced technique called IDPF [1] has been introduced. Which is constructed from the information based on Border Gateway Protocol (BGP) route updates and is organized in network border routers; on the other hand, this technique doesn't have the power to handle subnet spoofing address. Consequently, a new technique called Hop-Count Filtering (HCF) [6] proposed another simplified method to make out packets whose source IP addresses be spoofed. The information of source IP address and the consent hops from a server (victim) are stored in a table at the server side when there are free of attacks. When an alarm of attack is raised, the victim will check the incoming packets' IP addresses (source) and their responding hops to distinguish the spoofed packets. This is not compulsory for routers to work together mutually in the prescribed scheme; even though, it is hard to ensure the accuracy and integrity between the source IP addresses and their responding victim's hops. Furthermore, this proposed method is subjected to fixed spoofing or subnet spoofing. In addition, this method needs a huge memory to build a rounded table because of the vast 32 bits IP address space. Storage space be able to saved by aggregating or exploiting, but further occurrences are required to locate or query.

The bloom filter method has been used for defenses from TCP SYN flooding which is a popular DDoS attack technique [5]. It requires only a small amount of memory without rising the processing time. However, there are several parameters configured physically and this method is not suitable to network changing. At the same time as, it is deployed via routers and creates the hash operation on source address and destination address. It is critical to implement and attain a good result. Peng et al. [7] proposed a packet-filtering method based on the feature of source IP address spoofing on historical packets information. This method is specifically applied on the ingress routers. Usually, an ingress router maintains all the legitimate source IP addresses' history records, which have earlier appeared in the network. The historical record is used to decide whether to deny or admit an incoming packet, while the ingress router is overloaded. Moreover in all the cases, a router is associated with a large number of hosts, and attackers may penetrate filtering algorithms using low attack packet rates. Again, it is proved that the method is hard to obtain a good filtering result.

Previously stated methods are work with the DDoS characteristics of randomly spoofing source addresses. The skilled attackers might be conducted serious attacks with fixed spoofing or subnet spoofing.

In the recent DDoS attacking, some attackers exploit a large number of zombies in a great botnet to initiate attacks, so it is very difficult to find the real attackers, even if the zombies bare their position information. So far, the statistics-based filter methods ALPi [10] and PacketScore [8] observed the problem, and statistical method is proposed to cope with it at the possible victim site. The range of classic traffic attributes is accumulated by the victim. Generally, as a server, the victim make out a DDoS attack by comparing the profile of current traffic with the normal traffic, this is according to the current event. Based on the leaky barrel theorem or Bayesian probability to make decisions, it's using a scoring mechanism. Yet, the statistics-based method is more complex, because there are number of features involved. Concurrently, these methods employ the characteristic of bursting; but they are not able to filter low rate attacks, despite of the categories of spoofing address.

In this paper, based on the source information we proposed a novel attack packets filtering method, which extract the main advantages of the previous work. In an IP packet, the TTL value and source IP address are treated as information sources. The port number of a packet is not treated as information of data source, because most of the servers not fixed the source port numbers as victims. References [2], [4], and [6] have ensured that there is a strong association between the source IP addresses and the TTL values for normal traffic and the current IP address space is divided in nature. Our proposed methodology is to build a table based on the data of source IP addresses under hops from that source IP addresses to the server in standard condition. During the DDoS attacking period, the random spoofing of attacking packets would be filtered for their source IP addresses. In normal conditions, their hops would not deal with mapping relations and the no spoofing or subnet spoofing of attacking packets would also be filtered out for their statistics profiles flooded their gathered knowledge of the standard traffic. So as to reduce the storage space size, an advanced counting bloom filter is proposed to save the statistics of source IP address under the assured hops. The proposed method is referring as source informationbased filter (SIBF). SBF is installed at victim without the support of routers, and is suitable to put into practice.

Compared with the previous methods, the proposed SIBF methods possess the following features and also provide the novel contributions to the battle against DDoS attacks,

- SIBF works on the possible victim side, it has a strong incentive to apply the filtering function, and moreover, no collaboration among routers is needed.
- we deploy very restricted information, source IP addresses and hops, to filter attacking packets, it simplifies the requirements for implementation.
- The utmost pressure of storage space is reduced by the SIBF and accelerates the information retrieval processes considerably.
- Proposed method is outstanding against various IP address spoofing, which is so far the main method of DDoS attacks.

Source Information Based Filter Method

Method Analyzing

The observation shows that clients for a particular server are comparatively stable. Jung et al. [9] point out nearly 82.9% of all IP addresses in predicted flash crowd measures have sent a request before; conversely, only 0.6-14% of IP addresses in a Code Red attack had occurred. As per the statistical analysis of Internet traffic collected in a average size stub network, it is found that a great proportion of IP addresses emergence in Internet traffic consistently re-emerges based on daily observation. This implies that a consistent feature for recognizing legal traffic is that it has emerged at the website frequently. In the meantime, the hop from a client to server is relatively fixed. Based on this method, a server can launch its own clients IP address space under certain hop for every client, respectively. When hackers randomly spoof the source IP addresses and deliver the packets to the server (victim), so, the possibility is low that the spoofed IP address establishes in the particular clients IP address space under a specified hop. Proposed method is stimulated by these analyses. On the other hand, holding this information by establishing a table might be very expensive. Because it engages servers' massive memory spaces, and it also slowly carry outs

information retrieval in a enormous table. Hence, to solve these problems, bloom filter method is introduced and implemented [5]. Proposed method deployed at victim as an alternative of router, in which the bloom filter is just build with source addresses, which replace the pair of source address and destination address, the result would be improved compared with the scheme discussed in the paper of Chen et al. [5].

Bloom filter method is possibly to produce a bit of false positive rate, even though it decreases quickly the requirement of memory. Bloom filter method was proposed by Bloom [14] in 1970. Currently, the method has been adapted for use in some schemes for defending against DDoS attacks [11, 15]. Counting Bloom filter is the variation of normal bloom filter by adding function of counting. For the benefit of lightening the computing load of SIBF at victim, partial counting Bloom filter is the improved method of the counting Bloom filter method. The methodology in SIBF is given below.

The IP addresses are divided into *n* segments (n=1,2,3,4…32), size of each segment is 32/n bits, we look upon the segment as Part_IP, it has the value from 0 to $2^{32/n}$. Generally, while receiving a packet, the SIBF in a victim counts n Part_IPs of source IP packet, easy to get the all Part_IPs normal statistics profile of lawful source IP addresses. At the time of attacking, as per the theory of Bloom filter, if the consequent statistics value of one of the packet's Part_IPs received was 0, then the packet would be the attacking packet, and it should be discarded. At this juncture, the hashing function turn into the direct mapping it acquires a little CPU load of victim. Besides, the technique is useful to make out the attacks of subnet spoofing.

Suppose the attackers exploit the legal IP addresses to spoof the attacking packets' source IP addresses, bloom filter couldn't filter these packets fine. Hence, we were cleared from the thought of HCF [6], hops are introduced to separate the source IP addresses. The hops distribution of server's clients takes on certain statistics for a special server, laws [6]. At the time of attacking, a lot of zombies send attacking packets to server, counting number of relating hops of these packets increases by a great amount. When the statistics of a hop-count value is starting to exceed the normal statistical profile, sources of attacking fabrication of the hop in the distance will be deduced. While receiving a packet, its hop-count according to HCF is first calculated, and accumulates the counts of hop-count. After that, partial counting Bloom filter was constructed as stated earlier. During this way, spoofing of legal address problem can be solved, for various zombies to victim have the various hop-counts.

When the victim is free of attacks [8], every arriving packet's TCP/IP attribute values of a victim has a nominal profile. The traffic profile is constructed using both the Part_IPs from source IP address and hop-count coming from TTL attributes. Like to Packet Score, the current profile is comparing with the nominal one, SIBF is also able to differentiate valid packets from DDoS attacking packets of fix spoofing or subnet spoofing, even no spoofing. We just make use of two attributes relative to Packet Score, and relieve the load of victims.

Method Implementation

According to the statistics of source IP addresses an information table is build in case of hop-counts shifting between clients and its Internet server. The given IP address is divided into *m* segments, such as *P0, P1, ... Pm-1, 1≤m≤32*, and the range of every segment is *0* to *M-1, M=*$2^{32/m}$. According to reference [5], it is known that the hops from one node to another on the Internet are able to be calculated by TTL, the maximum hop is 31. *HCR* is denoted as the hop counter, so, *0≤HCR≤31*. The information for each of the segments on each hop value is accumulated, respectively for a given server.

The statistical value of the values i of the jth segment of the source IP is denoted as a^n_{ij}, which is written address, where $0 \le j \le m-1$, $0 \le i \le M-1$. In order to make the bloom filters problem clear, i.e., different hops are the same form, in the rest of this paper we only focus on one of the bloom filters, and thus, we omit the *n* in the variables. Therefore a^n_{ij}, $a_{j,i}$ for short. Actually, $a_{j,i}$ includes two parts, that is

$x_{j,i}$, $y_{j,i}$, wherein $x_{j,i}$ represents current statistics, and $y_{j,i}$ represents normal profile. In regular situation, a victim adjusts every so often its value of $y_{j,i}$ according to $x_{j,i}$, these are accumulated by the source IP addresses coming various authorized clients.

When attack occurs, then the evolution will be suspended. If a DDoS attack alarm is raised, then we check each incoming packet P' according to it hops, and chop P' into n parts as then P= { P_0, P_1, ….,P_{m-1} }. For part $P'j$, we can get the score of the part as follows:

$$f(P'_j) = \begin{cases} x_{j,k} - y_{j,k} & x_{j,k} > b_{j,k} \\ \infty & y_{i,k} = 0 \\ 0 & x_{j,k} < y_{j,k} \end{cases} \quad (1)$$

Here k is the value of $P'j$, the following norm is applied as a metric to differentiate the attack packet and legitimate packet by identifying the incoming packet P'

$$\| g(p') \| = \sum_{j=1}^{n-1} f(p'_j)_i \quad (2)$$

$\| g(p') \|$ is compared with a given threshold δ. Suppose the following holds, and then it is an attack packet.

$$\| g(p') \| \geq \delta \cdot z \quad (3)$$

Less z signifies severe attacks. It has somewhat to do with the worldwide statistics of the hop-count's packets. When hop-count equals to n, then set x_n as the current profile and y_n as the nominal profile, we have $z = y_n/x_n$. If z is 0, it can be concluded that there is no any legal packet come before, and then we consider the arriving packets with the hop-count are attacking packets. Suppose z is bigger than 1, it shows that the arriving packets are the legal packets for being out of attacks level less than its hop count.

According to the equation (4) the threshold δ can be drawn as follows:

$$\delta = m \max_{i=0,\, j=0}^{m-1\, M-1} | x_{j,i} - y_{j,i} | \qquad (4)$$

It seems that, δ is m times of the maximum of statistics' changing in some hop_count. For example, if $x_{i,j} > 0$ and $y_{i,j} = 0$, it shows that the packet with an IP address which not at all appears before, therefore, $||g(P')|| = \infty$, and it is definitely an attack packet. This results for the initiative of Bloom filter.

Algorithm and System Analysis

Requirement of Memory

Source IP address is divided into m Part_IPs, and counts them separately using SIBS. Thus the required space units can be obtained as $b = m*2^{32/m} *b$ which will decrease sharply with *m* increasing. To improve the filter effect, the filter will be build under each hop count. However it increases the need of memory. The exhaustive analysis of the storage space consuming in the proposed method is given now.

Assume that there is legal source IP address under a hop; now we set the size of the space units is b. As said earlier, each space unit needs saving nominal profile and current profile. The series of its value occupies the counting interval. Generally, it is sufficient to have 2 bytes of space to mark the value for a standard server. Hence, the essential space of SIBF is

$$men(n) = n \cdot b \cdot (2+2) = 4n \cdot m \cdot 2^{32} / m \ (bytes) \qquad (5)$$

Where n is the number of hops count and the number is less than 32, i.e., from 0 to 31. Often, the number of hops count of the arriving packets at a server is amongst from 10 to 20. In view of n may be different in every server, dynamically SIBF applies the space for storage under a hop count, and keep away from the waste of memory usage. For example, if n is 4 and m is 16, then the storage space is 64 KB. Suppose we change n as 3, the required space is 384 KB in other related conditions. Compare with HCF, SIBF occupies very less space than HCF while the memory is statically allocated; if dynamic tree structure applies in HCF to store data, then they use more or less the same space, however in the same situation HCF spend extra time querying data.

Efficiency of Filter

The false positive and the false negative represent the efficiency of a filter, where the false positive is the probability of identifying the legal packets as attack packets, and the false negative shows the probability of identifying the attack packets as legal packets.

In the proposed method, under some hop count, if L is the set of legal source IP addresses, then the number of entry of the set is l. Thus the sets of its n Part_IPs are $L_0, L_1, \ldots L_{n-1}$, which includes the entries of $l_0, l_1, \ldots l_{n-1}$. As the randomly spoofing attacks are coming, the probabilities of every segment of the packets is located on the same segment of normal packets are $l_0/M, l_1/M, \ldots l_{n-1}/M$. So, under randomly spoofing situation, the false negative is:

$$P_{fn} = \prod_{i=0}^{m-1} (t_i / N) = (\prod_{i=0}^{m-1} t_i)\ 32 / 2 \qquad (6)$$

As per equation (6), P_{fn} is part with the product of Part_IP's allocation of legal packets' source IP

addresses. For the reason that SIBF isolates source IP addresses to varied fields based on their hops, and also considering the district feature of IP, proposed scheme reduce the false negative more efficiently than other schemes.

According to the bloom filter theory, the false positive does not exist when all the normal packets have visited the server in beginning stage. Although in the real atmosphere, it is predictable that some clients access the server in the very first time during attacking periods. Based on the rule described earlier in this paper, suppose the clients are placed in where z>1 (attacking sources doesn't located the area of the clients), SIBF could not get the packets of those clients for attacking packets and filter them. In some extent, when there is no attacks source in whose area, the proposed method defends the new clients and decreases the ratio of false positive.

The situation becomes more complicated, to subnet spoofing. Suppose k segments is the fixed part which preceding the attack packets (k<m). If k is very small, for instance k=1, equation (6) can be used to estimate the false negative, now the equation become:

$$Sg_m = \prod_{i=k}^{m-1} l_i / M \qquad (7)$$

According to equation (7) the false negative will get bigger, if k is close to m. proposed method uses the scoring way to decrease the false negative. Among the attacks lasting, attack packets will acquire high score which lastly overrun the threshold to be filtered. Although the Part_IPs of subnet spoofing will attain high score and the legal packets have some of Part_IPs may be acquire so high score which are taken falsely for attack packets. So as to prevent this situation, the score rule of equation (1) is optimized to the follows

$$f'(p'j) = \min(\delta / q, f(p'j)) \qquad (8)$$

The upper limit of scoring is set by the equation (8), where q is a comparative factor i.e., configurable. Generally, it is set to n-1. When k is equal to q, the legal packets that have the same subnet of spoofing may be mistakenly pointed as attacks packets. Assume that the number of Part_IPs of subnet spoofing is $r_0, r_1, \ldots r_{k-1}$ correspondingly , when k>1, the false positive is

$$P_{fp} = \prod_{i=0}^{k-1} q_i / t \qquad (9)$$

It is easily observed that false positive is proportion to the Part_IPs distribution product of subnet spoofing to that of legitimate IP addresses. It is not possible for subnets spoofing to be excessive, therefore the false positive is restricted to low proportion.

All earlier analysis proved that the false probability of the proposed method is closely relate to the Part_IPs's distribution of legal IP addresses. The number of legal IP addresses range is

$$\max_{i=0}(l_i) \le l^n \le \prod_{i=0}^{m-1} l_i \qquad (10)$$

Apparently, with the increase of m, even if $l_0, l_1 \ldots l_{m-1}$ is constant, we can change the value of l in a substantial scale. So, the greatest impact of proposed method consists in the allocation of the segments of legal IP address, not its number. Though, the legal IP addresses can be separated with hop counts, and cut down the allocation product of the segments in the similar number of IP addresses.

3.3 Adaptive Analysis

Good filter to be in possession of nice adaptability is necessary. The network environment is frequently changing. If a filter requires configuring manually in varying situation to reach fine effect is a bad design. The adaptability of SIBF is discussed in the following.

The network keep relative stabilization in particular period, however the network topology will transform in a long period. The hops of certain client to the server would differ. Besides, in varying time, a server has differently nominal profile of the segments of source IP addresses of legitimate client. To adapt these changes, the equation (11) is applied to update the regular statistics profile.

$$y_{ji} = y_{ji\text{-}old} .\alpha + x_{ji} .(1-\alpha) \qquad (11)$$

The process works in regular interval. Here α is a correlation factor, $0<\alpha<1$, it is determined by the interval and the changing reliability of the clients of a server. When the number of clients or the hop of a client to a server differs, the corresponding x_{ji} would decrease to zero. Finally, the nominal profile y_{ji} would reduce, even drop to zero, changing adaptively. At the time of attacking, y_{ji} stops updating to carry out filtering and scoring; when attacks stop, x_{ji} is set as 0, and the updating process works again.

Performance Evaluations and Algorithms

Performance Evaluations

The DDoS attack system prototype has been established to evaluate the proposed method, for example, the most well-known DDoS attacks SYN Flooding, is employed in the proposed method. The cusum algorithm [15] is used for attacking detection, and at the victim side this method works as the packet filter. In order to compare the same, the 32 bits strict filtering algorithm of HCF is implemented.

Primarily, the false negative and the false positive status are examined, when the number of segments differs. The simulation of number of clients is kept at a permanent level of 15000. The Fig. 1 shows the results.

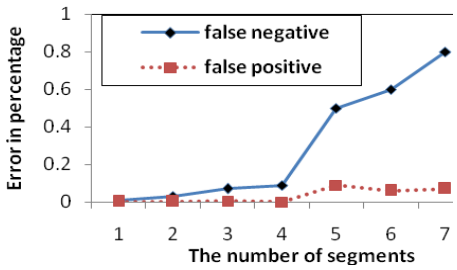

Fig. 1. Errors versus number of Segments

The simulation result shows that both false negative and false positive errors are relatively low (i.e., not greater than 3%) when the number of segments is not greater than 4 and the false negative increases significantly with the number of segments grows. With the growing of the number of segments, the percentage of the instances in a segment to the segment's space increases, so as per the equation (4) the false negative rises. The false positive remains stable, because it is related with the number of legal clients who not at all access a server before, it is independent from the number of segments.

Fig. 2 and Fig. 3 show the relation between the number of clients and error ratio, comparing with HCF algorithm, in the case of random spoofing. When n is equal to 3, based on the proposed method the result illustrates the SIBF is similar to HCF. The false negative increases relevantly, when the number of clients rising. Actually, the false negative is closely related with the client distribution to the server. When the number of legal clients increases, then the false positive slowly increase, since more number of clients maybe absent in the learning procedure of the proposed method.

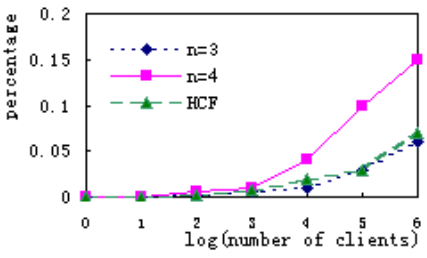

Fig. 2. False Negative with IP Random Spoofing

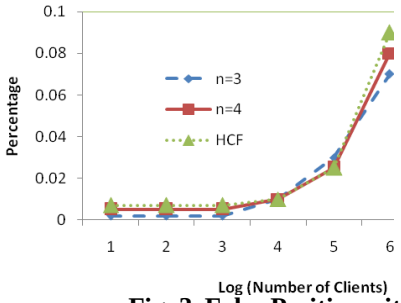

Fig. 3. False Positive with IP Random Spoofing

Hence, the impact of different methods with subnet spoofing proves the dissimilarity. The deployed attack packets are producing randomly in some subnet of C class. Fig. 4 and fig. 5 show the results. The figures disclose that the proposed method improves apparently the false negative; in the false positive concern, the proposed method is not better than HCF. The most important reason is that the scoring way influences the legitimate IP addresses those are very identical to subnet spoofed. Meanwhile, it is finding that the efficiency of filtering of the proposed method is connected with time (instance). The false positive continually increases, and false negative decreases for a while to become steady during the various time passes.

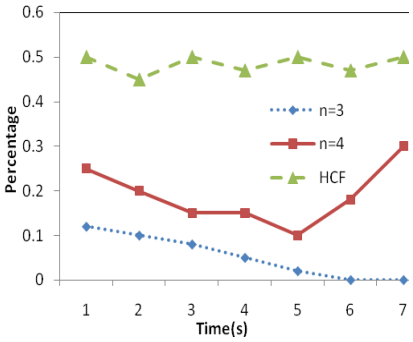

Fig. 4. False negative with IP subnet spoofing

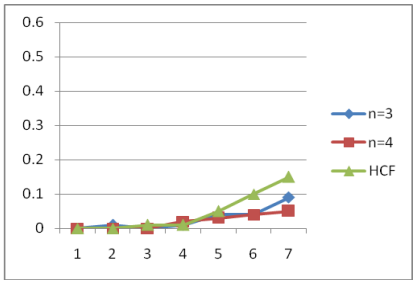

Fig. 5. False Positive with IP subnet Spoofing

Moreover, the attack intensity is automatically accommodated by the proposed method. All the incoming packets are added up for one hop of n. The attack intensity w can be calculated, in terms of the normal statistics of y_n and current statistics of x_n. Using equation (3), SIBF decreases the false and improves the defense effect for the various hops have different attack intensity.

Finally, the method is capable of equivalent for the change of the way of attacks. It is previously analyzed that SIBF adapts to the attacks of subnet spoofing and random spoofing. Besides, suppose the attackers bogus the TTL values to mislead one by getting error hops, as well the proposed method would get a good outcome. It is proved the profile of the hop could not alter by the attackers, as set onwards;

according to the regularity stated previously the attack packets would be filtered.

Filter Algorithms

Based on the results it can be concluded that there are two situations, when there is free no attacks it is consider as first situation, i.e., normal condition, , we present the algorithm of SIBF as follows,

1. Hop count *n* is calculated from the TTL field of the received packet.
2. Apply the space of $4.m.2^{32/m}$, when there is no statistics under the hop, bytes. Set all of x_{ji} to 0 and set x_n to 0, when attacking stage has just end.
3. Statistics is carrying out in terms of the segment values of the IP address.
4. Updating operation by equation (11) will be made, when the updating time expires.

Table 1 Attack free cases algorithm

But in the abnormal condition, i.e., when attack occurs, the filter algorithm is implemented in a different way.

1. Hop count *n* is calculated from the TTL field of the received packet.
2. Discard the packet as a attack packet and be over, when the normal statistics y_n is 0,
3. Statistics is carrying out in terms of the segment values of the IP address.
4. If $y_n > x_n$, be over
5. According to the equation (3) the packet will be scored. The packet is discarded, when the equation (3) holds.

Table 2 Ongoing attacks algorithm

Conclusions and Future Work

Based on source information statistics, a novel DDoS attack packets filtering algorithm is proposed in this paper. Server's clients source IP addresses and responding hops (i.e., source information) are analyzed by the method and follows some statistical patterns; though, it is very difficult for the spoofed packets to match the patterns. It accumulates the source IP addresses and hops, during the period of non-attacking, and therefore when the server is under attack it can easily differentiates the attack packets. Rather than treat the source IP addresses as a whole, these are divided into *m* segments. The proposed method offers high efficiency of diversely spoofed packets filtering when compared with the earlier works. Besides, the proposed works autonomously at the possible victim side, and in this method there is no cooperation among routers are needed. The results show that, the proposed method releases the challenge on storage space and speed of information retrieval at victims. Explore

The works can be explored in future as Neuro Fuzzy based Cluster formation for DDoS attacks detection with the statistical characteristics of data traffic. The load-demand capacity of ISP server at lean and heavy traffic times are observed to provide better detection of DDoS attacks. And also the storage structure can be improved to reduce memory requirement.

References

[1] Zhenhai Duan, Xin Yuan, Jaideep Chandrashekar. Controlling IP Spoofing through Interdomain Packet Filters. IEEE Trans. On Dependable and Secure Computing, Vol. 5, No. 1, JANUARY- MARCH , pp.22-36.(2008)

[2] Abraham Yaar, Adrian Perrig and Dawn Song, " StackPi: New Packet Marking and Filtering

Mechanisms for DDoS and IP Spoofing Defense", IEEE Journal On Selected Areas In Communications, vol. 24, no. 10, Oct. pp.1853-1863.(2006)
[3] K. Park and H. Lee, "On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets," Proc. ACM SIGCOMM, Aug. (2001).
[4] Fu-Yuan Lee, Shiuhpyng Shieh, "Defending against spoofed DDoS attacks with path fingerprint", Computers & Security, 24, pp.571-586.(2005)
[5] Wei Chen, Dit-Yan Yeung. Defending Against TCP SYN Flooding Attacks Under Different Types of IP Spoofing. in Proc. IEEE ICNICONSMCL, (2006).
[6] Haining Wang, Cheng Jin, and Kang G. Shin. Defense Against Spoofed IP Traffic Using Hop-Count Filtering. IEEE/ACM Trans. On Networking, vol. 15, no. 1, Feb, pp.40-53.(2007).
[7] Tao Peng, Leckie, C. Ramamohanarao, K. "Protection from distributed denial of service attacks using history-based IP filtering", ICC '03. May , Vol.1, pp: 482- 486.(2003).
[8] Yoohwan Kim, Wing Cheong Lau, Mooi Choo Chuah and H. Jonathan Chao, "PacketScore: A Statistics-Based Packet Filtering Scheme against Distributed Denial-of-Service Attacks", IEEE Trans. On Dependable and Secure Computing, vol. 3, no. 2, Apr, pp.141-155. (2006).
[9] Jaeyeon Jung, et al. "Flash crowds and denial of service attacks: Characterization and implications for cdns and web sites". WWW10, WWW2002, May 7-11, Honolulu, Hawaii, USA (2002).
[10] Paulo E. Ayres, Huizhong Sun, H. Jonathan Chao, Wing Cheong Lau, "ALPi: A DDoS Defense System for High-Speed Networks", IEEE Journal On Selected Areas In Communications, vol. 24, no. 10, Oct, pp.1864-1876.(2006).
[11] S. Abdelsayed, D. Glimsholt, C. Leckie, S. Ryan, and S. Shami. An efficient filter for denial-of-service bandwidth attacks. In IEEE Global Telecommunications Conference(GLOBECOM' 03), volume 3, pages 1353—1357, Dec (2003).
[12] Sertac Artan, N. Sinkar, K. et al. Aggregated Bloom Filters for Intrusion Detection and Prevention Hardware. Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE, Nov. pp. 349-354 (2007).
[13] Haining Wang, Danlu Zhang, Kang G. Shin, "Detecting SYN Flooding Attacks", in Proc. IEEE INFOCOM, pp.1530-1539 (2002).
[14] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. Communications of the ACM, 13(7):422—426, July (1970).
[15] E. Chan, H. Chan, K. Chan, V. Chan, S. Chanson, and etc. IDR: an intrusion detection router for defending against distributed denial-of-service (DDoS) attacks. In Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks 2004(ISPAN' 04)., pages 581—586 (2004).