Quality based Bottom-up-Detection and Prevention Techniques for DDOS in MANET

Laxmi Bala[1], A.K. Vatsa[2]

[1,2]Faculty of Engineering and IT, Shobhit University Meerut, UP, INDIA

Abstract

The distributed denial of service attack(DDoS) is a major threat to current internet security in MANET. Although the DDoS mechanism is widely understood, its detection is a very hard task because of the similarities between normal traffic and useless packet, sent by compromising host to their victims. Quality reducing attack is a new style of Distributed Denial of Service (DDoS) attack. The goodput and delay performance of TCP or UDP flows are very sensitive to such Quality reducing attacks. . In this paper a bottom up detection and prevention techniques for DDoS in MANET has been proposed thereby achieving an efficient quality of services provisioning. Our method relies on the use of monitoring and measurement techniques to evaluate the impact of SYN flooding attacks.

Keywords:

MANET, DDoS attack, TCP SYN flood attack, TTL.

1. Introduction

MANET is a distributed system that comprises wireless mobile nodes that can freely and dynamically self-organise into arbitrary, temporary, and ad hoc network topologies, allowing seamless interconnections without pre-existing communication infrastructure and central administration. Due to its unique characteristics, MANET is vulnerable to various security threats, and it is particularly susceptible to the DDoS attack. In the past few years, organizations have reported a growing number of incidents involving groups of attackers trying to damage commercial and institutional web applications by exhausting their resources through distributed denial-of-service(DDoS) attacks. Attacker groups understand that preserving application availability is a high priority for most organizations because availability influences application revenue, and therefore any reduction in the quality of service can reduce revenue as well as damage the organization’s reputation. A typical DDoS attack is the flooding attack in which attackers paralyse the target(computers or networks) by flooding excessive volume of traffic to deplete key resources of the target. In terms of DDoS attack methods the major ones are ICMP Flood attack, TCP SYN Flood attack and UDP Flood attack. To mitigate TCP SYN Flood attack in MANET, we propose to design a detection algorithm in this paper.

Quality reducing attack is an important DOS attack in Wireless Networks. The DDoS flooding attacks are characterized by the high rate or high volume. Recently a new attack called the shrew attacks or quality reducing attacks has been identified. Quality reducing attacks gradually reduces Quality of Services to end systems by strangling the TCP throughput heavily instead of entirely refusing the clients form the services. Instead of limiting its steady state capacity, quality reducing attacks targets the systems adaptive behavior. Source and destination IP spoofing are used by quality reducing attacks. Due to the absence of dissimilar periodicity, the packets are not filtered accurately. Quality reducing attacks are commenced through multiple zombies and spoof header packet information so that they can escape from trace back techniques. In fact it is necessary to control the frequency domain characteristics of attacking flows. The attacking period has to be close to the Retransmission Time Out (RTO) so that TCP flows are efficiently strangled. Though the source IP addresses of the packet header are falsified, the malicious flow detection mechanisms are relinquished by energy distribution pattern using traffic spectrum [4].

TCP SYN Flood attack is based on the exploiting of standard TCP three-way handshake. Once a server received an initial SYN request from a client, it sends back an SYN/ACK packet and waits for the final ACK packet from client. However, it leaves server system waiting for the non-existent final ACK packets. Considering that the server only has a limited buffer queue for new connections, SYN Flood causes the server to be unable to process other incoming connections as the queue gets overloaded. A half-open connection is a connection state in which the server is waiting for the acknowledgement ACK from a client. This state is normally caused by an uncompleted TCP three-way handshake. In such a case, the server will try to complete the three-way handshake by resending SYN/ACK packets. The object of this is to minimize the damage caused by network congestion and to improve the reliability of the three-way handshake.

Normal half-open connections are half-open connections caused by network congestion or other network errors. Abnormal half-open connections are those which can be observed on a victim server during DDoS attacks (e.g., a SYN flooding attack).The key problem is to distinguish the abnormal half-open connection from the normal half-open connection so that the abnormal connection can immediately be released and ceases to consume server

Manuscript received August 5, 2013
Manuscript revised August 20, 2013

resources. As noted, most normal half-open connections arise from network congestion whereas abnormal half-open connections have no relevance to the short traffic delay that is seen between network routers in a normal environment.

Network traffic congestion can be inferred from features such as increased packet delay, a high packet loss ratio, and a near-capacity queue at a congested router. If these signs can be detected, a given half-open connection is most probably caused by a traffic congestion and is therefore a normal half-open connection. If these signs are not present, a given half-open connection is regarded as an abnormal half-open. As IP packets are routed across the Internet, the time-to-live (TTL) field is decremented. This field in the IP packet header is used to prevent packets from being routed endlessly when the destination host cannot be located in a fixed number of hops. It is also used by some networked devices to prevent packets from being sent beyond a host's network subnet.

This paper is organized into sections. Section–1, Section-2 contains introduction and background. Section 3 presents the first of the two proposed sequentially implemented components of our system, quality reducing methods by which an attacker can attack in victim server and second component bottom up approach for quality based detection scheme for SYN flooding DDoS attack. Section 4 presents window control and TTL-based rate limiting counteraction scheme for prevention of SYN flooding attack. Section 5 offers our conclusion and future scope.

2. Background

Reduction of Quality (ROQ) attack[4] is one of the Denial of Service (DoS) attacks which affect the MANETs. Instead of refusing the clients from the services completely, these RoQ attacks throttle the TCP throughput heavily and reduce the QoS to end systems. An AAT-based DDoS model (ADDoSAT)[2] is developed to assess the potential threat from the malicious packets transmission on the primary victim server and to facilitate the detection of such attacks; an AAT-based bottom-up detection algorithm is proposed to detect all kinds of attacks based on AAT modeling. Research[19] into victim-side defenses is encouraged by the fact that victims are more willing to deploy the resources to defend system against DDoS attacks. Indeed, the majority of autonomous defenses are set up on the victim side.

The reputation-based incentive mechanism[7] is effective in tackling DoS attacks that occur due to selfish and malicious nodes. The misbehaving node detection rate was higher when the aggregated reputation rating, as opposed to just neighborhood information, was used. The flooding attack considered in this work performs at the network layer. It aims to paralyze the entire network, rather than any particular node, by injecting overwhelming attack traffic (e.g. RREQ broadcasting) into the MANET. A flow[1] is defined as a set of packets that have same source and destination addresses, i.e.(SA,DA). We define that a node sees a flow when this node receives any packet belonging to this flow. An efficient router[5] can detect the SYN flood attacks. Every network should have one router in terms we have to design our network. Ever entry of packet should be monitor then check the IP address if it's legitimate then only it can allow to networks. If there is any IP spoofing technique happen in the IP header that packet will restricted. Using router we can detect the SYN flood attacks because SYN flood attacks happen after the packets came into the system by the unauthorized user. If we use router in every networks the earlier stage itself spoofed packets detected, it's very easy to solve the problem compare with after happen the attack. The approach adopts a novel mechanism to ensure detecting SYN flooding attack at its early stage. It is the fact[6] that the normal half-open connection maintained inside a server exists as a result of network traffic congestions while the half-open connections caused by a SYN flooding are launched only by attackers. To detect legitimate established connections[13], we take advantage of the fact that all segments originated from the server with the ACK flag set on and the SYN flag set off indicate a successfully established connection. In this case, the probability that the sampled packet contains one of multiple ACK segments coming from the server is greatly increased.

An active probing scheme[19] is used to diagnose the network traffic congestion status. We can quickly classify a half open connection as either normal or abnormal from the knowledge of network traffic distribution. It is possible to obtain the network traffic distribution either using our active delay probing method DARB (DelAy pRoBing), or using the traffic delay history .By sending a packet to the claimed host that will cause a reply we can check to see if the TTL in the reply is the same as the packet being checked. If they are of the same protocol, they generally have the same TTL. Because different protocols use different initial TTLs, when the probe packet is of a different protocol, we must infer the actual hop count [15]. The final TTL value when a packet reaches its destination is, therefore, the initial TTL decreased by the number of intermediate hops(or simply hop-count). 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 (OSes) 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 [16]. Some of our solutions depend on setting a TCP packet's time to live (TTL) value such that the packet will leave the peer's internal network, but not reach the buddy's NAT. For different networks this value will be different, and as such it must be able to be dynamically determined [18].

Tracing the paths of IP packets back to their origin, known as IP trace back, is an important step in defending against DOS attacks employing IP spoofing. The main idea behind packet marking is to record network path information in packets. In mark based IP trace back, routers write their identification information (e.g., IP addresses) into a header field of forwarded packets. The destination node then retrieves the marking information from the received packets and determines the network path [26]. In SYN cache, a hash table keeps track of the half open state connections instead of relying on the backlog queue provided for each application. SYN cookie eliminates the need for the backlog queue to keep track of each SYN request. In SYNDefender, the firewall intercepts the SYN request from the client and sends the SYN&ACK packet on the behalf of the server. In Synkill, source IP addresses are classified in a database as good or bad based on observed network traffic and administratively supplied input. Bad source addresses are sent the RST packet to terminate their requests while good ones are allowed to carry on with the handshaking [10].

3. Proposed Work

The proposed works are illustrated as following in section 3.1 and 3.2

3.1 Working Principle of Quality based Bottom-up-Detection and Prevention Techniques for DDOS in MANET:

- Firstly extract feature from a group of captured network packets within a specified time period. The extracted information includes packet protocol type, packet flag, source IP, destination IP, sequence number, acknowledgement number, TTL value etc.

Protocol Type	Flag	TTL	Sequence Number	Acknowledgement Number	Source Address	Destination Address	Window Size

Figure-2: Packet Format

- Server listen TCP request by receiving large number of packets. In which server receives any particular number of packets with SYN flag in TW. In this step firstly Server filters the SYN packets to record related information into R1, R2 and R3 record then response accordingly.
    - o R1 records first SYN packets information which is requesting for new TCP connection.
    - o R2 records that SYN packets information which has completed three way handshakes.
    - o R3 records other type of SYN packets information.

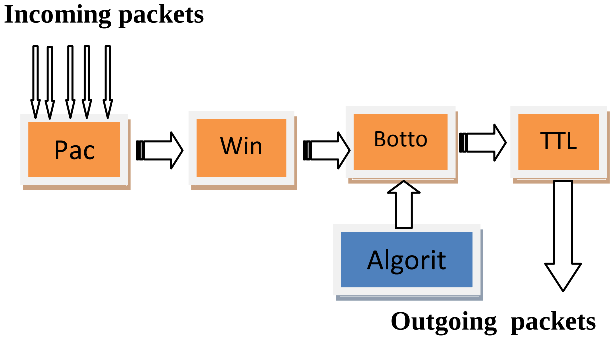

Figure-1: Architecture of Quality based Bottom-up-Detection and prevention Techniques for DDOS in MANET

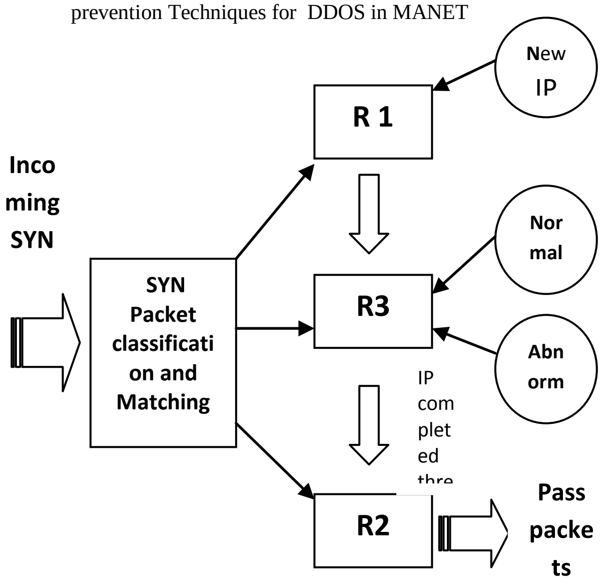

Figure-3: Classification of SYN packets

- Server response SYN packets whose record related information in R3. Then send large number of SYN/ACK packets. In which server sends any particular number of replies with SYN/ACK flag within TW.
- Server waits for ACK response from client within TW. If ACK is received so these TCP connection is completed and record its information in R2.
- Otherwise if there is no response packets are received, then half open connection state will occur and server again retransmit SYN/ACK packet to client 5 times, doubling the timeout value after each retransmission.
- If again server will not get ACK packet then calculate delay using DARB(delay probing method). We classify the all half open connection in as either

normal or abnormal from calculating average delay value (Davg) in a session.

- When the number of abnormal half opens exceeds a predefined threshold Th, an attack SYN flooding attack is detected and DDOS alarm message will sent out by the monitor node. At what level the threshold Th is set depends on how many half-open connections a server can tolerate. If the server reserves more resources for half-open connections, the threshold Th can be set accordingly larger.
- During the detection period, the monitor node analyzes and logs all the TCP connections incidents into records once a set of the captured data characteristics match signatures at any of the step of this bottom up approach. The records are examined recursively by this detection algorithm and the ultimate attack is identified once the root node is reached.
- Also, if the extracted data characteristics cannot match any signatures, those packets will be regarded as normal network traffic and the detection system ignores them. The whole detection process continues recursively till detection been terminated.

3.2 Mechanism based on bottom-up-detection and prevention techniques for DDOS:

The proposed mechanism is discussed as follows in three phases.

Phase–I: Quality Reduction Based Attacks

QRB_Attack( )

{

- When an attacking host send SYN(k) packet for new connection to victim server with its sequence number , ACK number, spoofed source IP address, destination IP address, initial TTL value, flag value, protocol type, window size packet information.
- The number of session in the server side is limited only by memory buffer and can grow as new connection arrive, but the client must allocate a random port before sending the first SYN packet to server. This port allocated during the whole connection.
- Victim server allocates memory for that host and sends SYN/ACK to that attacker consumes one sequence number and waits to receive for ACK from attacking host. This state is called half open connection state.
- Each half-open connection will remain on the memory buffer until it times out, it will retransmit the duplicate SYN/ACK 5 times, doubling the time-out value after each retransmission. The initial time-out value is 3 seconds, so retries are attempted at 3, 6, 12, 24, and 48 seconds.

```
ServerRetransmitSYN/ACKpacket()
{
    Timeout=t;
    A=0;
    ServerResponseTCPRequestSYN/ACKPacket( );
    While(A≤5)
    {
        For(i=1;i≤t;i++)
        {
            ServerWaitACKResponsePacket( );
        }
        ServerResponseTCPRequestSYN/ACKPacket( );
            //Server resends SYN/ACK to client
            A++;
            t=2*t;
    }
}
```

- More and more requests will accumulate and fill up the memory buffer at server side. SYN/ACK response packets do not reach the attackers machines due to spoofed IP address and the final ACK packet are not sent to the victim server to complete the 3-way handshake.
- Attacker send large number SYN packets with spoofed source IP for preventing services to be granted to other legitimate requests.
- Therefore, no new request, including legitimate requests, can be processed and the services of the system are disabled. SYN Flood attack is detected which causes server to be unable to process other incoming connections as the memory stack gets overloaded.

}

AlteredSequenceNumber( )

{

- Attacker use as like man in the middle attack using packet sniffers and reads TCP header.
- Knows the sequence number, ACK number, ports and protocol number excepted by the server.
- Attacker forges the packet and sends it to server before client does so.

}

GenerationSpoofedIPaddress( )

{

- The request to the server by client would be redirected though proxy server and Attacker can receive information from them.
- Or attacker set up a counterfeited Web site which looks exactly like the legitimate Web site, including setting up the web server, applying the DNS server name, and creating the web pages similar to the destination Web site, etc.

}

AlterTTLValue( )

{

- Client initiate TCP connection by sending out an initial SYN packet with the initial TTL set too low. The IP TTL field limits the lifetime of packets transmitted across the Internet and is decreased by each forwarding device (router).
- Once SYN packet is dropped on route, its TTL value is increased or decreased by Attacker and client waits round trip time for SYN/ACK packet from victim server.
- Attacker can decrease the TTL value by which its value reaches zero before arriving at the destination host , the router drops the offending packets and transmits an ICMP (Internet control message protocol) ‘TTL exceeded in transit’ error message to the original host, informing the original host of the packet’s timeout.
- If this SYN/ACK packet is not received within that round trip time by client then server resend duplicate SYN/ACK packet with double round trip time and the victim server may try again to discover a route by broadcasting another SYN/ACK packet up to a maximum of retry times at the maximum TTL value.
- Attacker can broadcast SYN packet in an incrementing ring to reduce the overhead caused by flooding the whole network. The packets are flooded in a small area (a ring) first defined by a starting TTL in the IP headers. After round trip time, if no SYN/ACK has been received, the flooded area is enlarged by increasing the TTL by a fixed value.
- The procedure is repeated until an SYN/ACK is received by the client for three way handshake and prevents the legitimate user to grant services from victim server.

}

Phase –II: Bottom- up Approach for detection of TCP SYN Flood Attack:

Note that we also define four threshold values T, N, Delay_{th}, Th in this approach. T represents the specified time window (TW), N represents the specified number of packets, Delay_{th} represents maximum packet delay time, Th is for maximum number of abnormal half open connection. It supports a basic list of signatures for every step. A signature is a set of match malicious network packets. We perform one or more actions on traffic that matches a signature.

Step-1 Server received large no. of SYN packets request from clients.

```
ServerListenTCPRequestSYNPacket( )
{
 IF(protocol.type = tcp && tcp.Flag = SYN(k)
 && destination.IP = victimServer.IP && t<=T
 && n<= N )
 {
   If(Server listen SYN packets for new TCP connection)
   {
        SeqNo=SeqNo+1;
        //consumes one sequence number
        Store SYN packets related information to R1;
        Drop SYN packets;
        Move this SYN packets information to R3;
   }
   Elseif(Server received SYN packets which record is in R2)
   {
        Pass SYN packets;
   }
   Else
   {
        Drop SYN packets;
   }
  }
}
```

Step-2. Server sends large no. of SYN/ACK packets to clients.

```
ServerResponseTCPRequestSYN/ACKPacket( )
{
 If(protocol.type = tcp && tcp.Flag =
 ACK(k+1)+SYN(j) && source.IP =
 victimServer.IP && t<=T && n<=N)
  {
   SeqNo = SeqNo +1;
  }
```

Step-3. Server waits for ACK packets from clients to establish TCP connection.

```
ServerWaitACKResponsePacket( )
{
  If(protocol.Type = tcp && source.IP = victimServer.IP && t<=T && n<=N)
  {
     If(tcp.Flag = ACK(j+1))
     {
            ACK is received;
            Store SYN packet related information to R2;
         }
         Else
         {
           ServerRetransmitSYN/ACKpacket( );
           tcp = HalfOpenConnection;
           probe(Input:HalfOpenConnection,Output: delay value(X));
         }
      }
      For all HalfOpenConnection in a session S
      {
         $Davg = \sum_{i\in S}^{\infty} Xi / |S|$ ;    /*calculate average
         delay value where X is delay value for ith half
         open connection*/
         If(Davg ≥ $Delay_{th}$)         // $Delay_{th}$ used as a threshold value
         {
          "Connection is normal HalfOpenConnection due to congestion in traffic";
         }
         Else
        {
          "Connection is abnormal HalfOpenConnection ";
           AH = AH+1;
         }
      }
```

Step-4. TCP SYN flooding DDoS attack detected.

```
SYNFloodAttack()
{
 If(AH ≥ Th)
{
   GenerateAlarmMessage()
  {
     "SYN Flooding Attack is detected"
   }
 }
}
```

```
/*Connection is Established
```

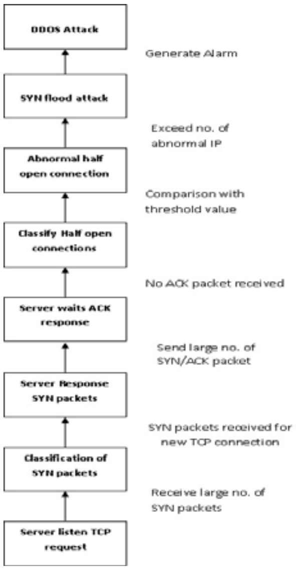
DDOS Attack
Generate Alarm
SYN flood attack
Exceed no. of abnormal IP
Abnormal half open connection
Comparison with threshold value
Classify Half open connections
No ACK packet received
Server waits ACK response
Send large no. of SYN/ACK packet
Server Response SYN packets
SYN packets received for new TCP connection
Classification of SYN packets
Receive large no. of SYN packets
Server listen TCP request

Figure-4:Bottom up detection techniques for DDOS attack

Phase–III Prevention

The prevention techniques are discussed as below.

1. Window-based Control for Normal Half Open Connection:

Window Control [28] is needed when simple rate control is not sufficient to police the traffic. Fixed resources which are based on capacity, as opposed to rates, are examples where window control is applicable. By enforcing a separate window for each resource, we can ensure that the traffic stays within the administrator-decided policies, and does not consume too much of a fixed resource. Examples of such resources are CPU cycles on end server, memory, network buffers (like sk buffs), and protocol state buffers (like SYN backlog queues). Windowing allows a resource to be self-regulated, as new requests cannot enter the system until the earlier requests have left the system.

In this approach we proposed a window limit per resource or per traffic aggregate. This allows us to control how a certain resource can be consumed by a traffic class at any given time. After this limit is reached, incoming requests or packets seeking this resource are dropped or delayed at the QOS regulator until the server sends some kind of indication that an earlier request from this traffic class has freed its resources. When this happens, more flows or requests can be admitted. The window limit quantifies the resource availability. The same resource can be shared across multiple different traffic classes through weighted fair sharing. This allows regulation of resource consumption of each class when the resource is in demand while allowing resource shares to be distributed across remaining classes when one traffic class is idle. For critical content, for example, when a transaction is going on at a web server, a portion could be allocated such that all transaction requests are guaranteed some percentage of resources even during overload. This ensures that critical transactions or preferred flows are not starved in presence of overload, or denial of service scenarios.

2. TTL- based Packet Filtering Approach for Abnormal Half Open Connection:

An entry of an IDS or firewall log file typically corresponds to a packet and includes the following information: source and destination ID (e.g. IP address in the TCP/IP context), the timestamp (e.g. when a packet is received) and etc. Our DDoS traffic analysis is mainly based on IP addresses and timestamp.

In abnormal connection there is two type of IP address-spoofing flooding attack is done. First is random address-spoofing flooding attacks in which each SYN packet sent out by each attack node is allocated a random pair of source address and destination address (SA,DA) and second one is fixed-address-spoofing flooding attacks, attackers cannot use randomly generated addresses but their own addresses for SYN packets. In other words, attackers can only use a fixed (SA, DA) pair for all SYN packets.

In this paper, we only consider the fixed address spoofing flooding attack. Internet paths are strongly dominated by a single route, and the routes of about two-thirds of them persist for days or weeks. This obviates the problem of forged IP addresses because attacking packets from the same attack source can be identified by the fact that they will have hopped across the same number of routers along their routing path.

When an Attacker packet is sent between two hosts, as long as the same route is taken, the number of hops will be the same. This means that the initial TTL will be decremented by the same amount. SYN Packets sent near in time to each other will take the same route to the destination. The result is that their TTL value will be the same upon arriving at a victim server. The central assumption of this is that attackers do not change the initial TTL value for each attacking packet.

We first determine the initial TTL value of a SYN 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 initial TTL value depends on different operation systems (OSs). Current OSs uses only a few selected initial TTL values, i.e., 30, 32, 60, 64, 128 and 255.

When abnormal half-open connections are detected, their TTL values are recorded in a table. When count of same initial TTL value reached to threshold M that time by packet filtering those SYN packets and limits the packet rate. Or we can use trace back technique [8] which will find out the source of attack traffic by tracing back the routers through which the attack packet has traversed and blacklist that attacker.

Blacklisting of the Nodes: These IP spoofing nodes having the attacking packet are set into the blacklist using the monitoring node. Nodes set into the blacklist are involved only in the data forwarding and is not able to perform any other operations.

Transmission security is based on digital signature method in which each node uses private key to sign the blacklist.

- The signed blacklist list is transmitted to the master node(MS) by each monitoring node.
- MS integrates all blacklisted nodes collected from the monitoring nodes.
- The node which is placed in more than a certain number of local blacklists is considered as an attacker.
- The attacker will be notified by the MS through the Notification message to all the monitoring nodes.
- All the monitoring nodes become aware of the attacker and block that node from further transmissions.

Filtering all packets having a certain TTL value would result in the filtering of legitimate as well as attack packets. Hence, our TTL-based rate-limit scheme includes rules for distinguishing normal from spoofed packets. It does this by observing TCP three-way handshake behaviors. During a normal three way handshake procedure, Syn(k), Ack(k + 1) + Syn(j) and Ack(j +1) can be captured at the victim side. However, during a spoofed TCP connection, whereas the first and second round handshakes can be identified while the third round handshake, Ack(j +1), cannot. On this basis, we conclude that a connection is legitimate if it is possible to capture its third round handshake Ack(j + 1). Traffic from this IP will not be confined within our rate-limit scheme.

4. Conclusion

In this paper, we focus upon the quality reducing based attacks in MANETs. Instead of refusing the clients from the services completely, the quality reducing attacks throttle the TCP throughput heavily and reduce the quality of services to end systems gradually. The quality reducing attacks may not filter the attack packets precisely. In order to avoid this, we presented the quality based bottom up approach for detection and prevention for DDoS attacks in MANET. Our approach can accurately identify the SYN flooding DDoS attack and consequently applying window control to reduce congestion and TTL based packet filtering technique to identify attacker and blacklist that attacker. In this paper we use delay as a sign of congestion then analysis the difference of half open connections originated from DDoS attacks and normal traffic congestion.

5. Future Scope

This paper we analyze the TTL value for calculating delay of packet which mostly effect the quality of services. When an attacker forges the packet and sends it to server before client does so by knowing its TCP header information like sequence no., ACK no., etc. The request to the server by client would be redirected though IP spoofing and Attacker can receive information from them. If there is congestion during attacks, by increasing TTL value of packets, then our method cannot detect these malicious packets. If these assumptions hold, the described methods may result in false negative, that is, invalid packets may not appear to be spoofed. The detection algorithm needs the capability to detect any newly starting attacks not relating to the current happening ones.

References

[1] Yinghua Guo and Ivan Lee, “Forensic analysis of DoS attack traffic in MANET”,Fourth International Conference on Network and System Security,2010.

[2] Jie Wang, Raphael C.-W. Phan, John N. Whitley and David J. Parish, “Augmented Attack Tree Modeling of Distributed Denial of Services and Tree Based Attack Detection Method”,10th IEEE International Conference on Computer and Information Technology ,2010.

[3] Rachana Yogesh patil and Lata Ragha, “A Rate Limiting Mechanism for Defending Against Flooding Based Distributed Denial of Service Attack”, World Congress on Information and Communication Technologies, 2011.

[4] S.Venkatasubramanian and N.P.Gopalan, “A Flow Monitoring based Distributed Defense Technique for Reduction of Quality Attacks in MANET”, International Journal of Computer Applications, Volume 21, No.1, PP 0975 – 8887, May 2011.

[5] S.Gavaskar, R.Surendiran and Dr.E.Ramaraj, “Three Counter Defense Mechanism for TCP SYN Flooding Attacks”, International Journal of Computer Applications ,Volume 6, No.6, (0975 – 8887), September 2010.

[6] Bin Xiao, Wei Chenz, Yanxiang Hez and Edwin H.-M. Sha, “An Active Detecting Method Against SYN Flooding Attack”,

[7] Mieso K. Denko, “Detection and Prevention of Denial of Service (DoS) Attacks in Mobile Ad Hoc Networks using Reputation-Based Incentive Scheme”, Jouraral Systematics, Cybernatics and Informatics, vol 3, N0.4.

[8] Jeevaa Katiravan, C. Chellappan and J. Gincy Rejula, “Detecting the Source of TCP SYN Flood Attack using IP Trace Back”, European Journal of Scientific Research ISSN 1450-216X Vol.71 No.1, pp. 78-84,2012.

[9] Neeraj Sharma, B.L. Raina, Prabha Rani, Yogesh Chaba and Yudhvir Singh, “Attack Prevention Method Methods for DDOS Attack in MANET”, Asian Journal Of Computer Science And Information Technology1:1, 18 – 21,2010.

[10] Vrizlynn L. L. Thing, Morris Sloman and Naranker Dulay, “Enhanced TCP SYN Attack Detection”.

[11] Wei Chen and Dit-Yan Yeung, “Defending Against TCP SYN Flooding Attacks Under Different Types of IP Spoofing.”

[12] L.Kavisankar and C.Chellappan, “CNoA: Challenging Number Approach for uncovering TCP SYN flooding using SYN spoofing attack.”

[13] Maciej Korczy´nski, Lucjan Janowki and Andrzej Duda, “An Accurate Sampling Scheme for Detecting SYN Flooding Attacks and Portscans.”

[14] Haining Wang, Danlu Zhang and Kang G. Shin “Detecting SYN Flood.”

[15] Steven J. Templeton and Karl E. Levitt, “Detecting Spoofed Packets.”

[16] Haining Wang, Cheng Jin and Kang G. Shin, “Defense Against Spoofed IP Traffic Using Hop-Count Filtering”, IEEE/ACM Transaction on Networking, Vol.15, No. 1, FEBRUARY 2007.

[17] Nallamala Sri Hari, N. Srinivas Rao and N. Satyanarayana, “A Novel Routing Attack in Mobile Ad Hoc Networks ”, Indian Journal of Computer Science and Engineering, Vol. 1 ,No. 4, 382-391.

[18] Andrew Biggadike, Daniel Ferullo, Geoffrey Wilson and Adrian Perrig, “NATBLASTER: Establishing TCP Connections Between Hosts Behind NATs”.

[19] Bin Xiaoa, Wei Chenb and Yanxiang He, “An autonomous defense against SYN flooding attacks: Detect and throttle attacks at the victim side independently”, J. Parallel Distributed Computing, 68 , 456 – 470, 2008.

[20] Qiming Li Temasek, Ee-Chien Chang and Mun Choon Chan “On the Effectiveness of DDoS Attacks on Statistical Filtering”.

[21] Shashank Lagishetty, Pruthvi Sabbu and Kannan Srinathan, “DMIPS - Defensive Mechanism against IP Spoofing”.

[22] Changhua Sun, Chengchen Hu, Yachao Zhou, and Xin Xiao,Bin Liu, “A More Accurate Scheme to Detect SYN Flood Attacks”.

[23] Jarmo Molsa, “Mitigating DoS Attacks against the DNS with Dynamic TTL Values”.

[24] S.A.Arunmozhi1and Y.Venkataramani “DDoS Attack and Defense Scheme in Wireless Ad hoc Networks”, International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.3, May 2011.

[25] Ruiliang Chen, Jung-Min Park and Randolph Marchany, “A Divide-and-Conquer Strategy for Thwarting Distributed Denial-of-Service Attacks”, IEEE Transaction on Parallel on Dstributed Systems, Vol. 18, No. 5, May 2007.

[26] G.Pradeep Reddy, A.Ananda Rao, “An Implementation Of Botnet Detection Algorithm For Grid Networks”, International Journal Communication & Network Security (IJCNS), Volume-I, Issue-II, 2011.

[27] Rodrigo Braga, Edjard Mota and Alexandre Passito,”Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow”, 35th Annual IEEE Conference on Local Computer Networks LCN 2010.

[28] Aman Garg and A.L.Narasimha Reddy,"Mitigation of DoS attacks through QoS regulation."

[29] Wei Ren, Dit-Yan Yeung, Hai Jin, and Mei Yang, " Pulsing RoQ DDoS Attack and Defense Scheme in Mobile Ad Hoc Networks", International Journal of Network Security, Vol.4, No.2, PP.227-234, Mar. 2007.

[30] Haining Wang, and Kang G. Shin "Transport-Aware IP Routers: A Built-In Protection Mechanism to Counter DDoS Attacks", IEEE Transaction on Parallel on Dstributed Systems, Vol. 14, No. 9, September 2003.

[31] Rizwan Khan and A.K.Vatsa, "Detection and Control of DDOS Attack over Reputation and Score Based MANET", Journal of Emerging Trends in computing and Information Sciences, Vol. 2,No.11,October 2011.

Laxmi Bala is pursuing M-Tech in Computer Engineering from Shobhit University, Meerut. She obtained B-Tech in Computer Science & Engineering from IIMT Engineering College, Meerut in 2009. She has been in teaching since last three years. Now days she is working as lecturer in ABSS Institute of Technology, Meerut. Her research interests are in MANET(Mobile Ad-Hoc network) and Network Security.

Avimanyou Kumar Vatsa is working as Assistant Professor and Coordinator - CSE at Shobhit University, Meerut, (U.P.), INDIA. He obtained his M-Tech (Computer Engineering) with Hons. from Shobhit University and B-Tech(I.T.) from V.B.S. Purvanchal University, Jaunpur (U.P.). He has worked as software engineer in software industry. He has been in teaching from more than one decade. During this short period of time, he has been supervised several dissertation of M.Tech. students. He is on the editorial board and reviewers of several international and national journals in networks and security field. He has been member of several academic and administrative bodies. During his teaching he has been coordinated many Technical fests and National Conferences at Institute and University Level. He has attended several seminars, workshops and conferences at various levels. His many papers are published in various national, international journals and conferences. His area of research includes MANET (Mobile Ad-Hoc network), Network Security, Congestion Control and VOIP-SIP (Voice over IP).