A New Approach to Detect, Filter And Trace the DDoS Attack

S.Gomathi,
M.Phil Research scholar, Department of Computer Science, Government Arts College, Udumalpet-642126.
E-mail id: gomathipriya1988@gmail.com

Dr.E.Karthikeyan M.Sc., M.Phil., Ph.D.,
Head & Assistant Professor,Department of Computer Science,Government Arts College, Udumalpet-642126.
E-mail id: e_karthi@yahoo.com

--ABSTRACT--

With the tremendous growth of network-based services and users of the Internet, it is important to keep the data and transactions in the internet more secure. Since the volume of sensitive and valuable information passing over the Internet is growing very large, the security attacks like Phishing, Spoofing, Flooding, Virus, and Spam are increasing. The Internet attackers can forge the source address of IP packets to both maintain their anonymity and redirect the blame for attacks. These spoofing packets are often part of some malicious activity, such as a DDoS attack. To thwart DDoS attacks, researchers have taken two distinct approaches: *packet filtering* and *packet tracing*. Packet filtering mechanism defines to detect and filter the attacked packet and Packet tracing mechanism defines to detect and trace the source, block the attacked traffic. In the proposed work, combining these two mechanisms to effectively detect, filter and also trace the DDoS attack.

Keywords: DDoS attack, Internet,IP spoofing, packet filtering, packet tracing.

1. Introduction

Today, the Internet is an essential part of our everyday life and many important and crucial services like banking, shopping, transport, health, and communication are partly or completely dependent on the Internet. As the Internet was originally designed for openness and scalability without much concern for security. Unfortunately, it is not possible to reliably determine the source of received IP packets, as the protocol does not provide authentication of the packet based on the source address field, which can be easily faked (IP spoofing). Furthermore the Internet routing infrastructure also does not keep information about forwarded packets. Malicious users can exploit these design weaknesses of the internet to wreak havoc in its operation. Incidents of disruptive activities which have raised the most concern in recent years are the denial-of-service (DoS) attacks [1] whose sole purpose is to reduce or eliminate the availability of a service provided over the Internet, to its legitimate users. This is achieved either by exploiting the vulnerabilities in the software, network protocols, or operation systems, or by exhausting the consumable resources such as the bandwidth, computational time and memory of the victim. The first kind of attacks can be avoided by patching-up vulnerable software and updating the host systems from time to time. In comparison, the second kind of DoS attacks is much more difficult to defend. This works by sending a large number of packets to the target, so that some critical resources of the victim are exhausted and the victim can no longer communicate with other users.

In the distributed form of DoS attacks (called DDoS), the attacker first takes control of a large number of vulnerable hosts on the internet, and then uses them to simultaneously send a huge flood of packets to the victim, exhausting all of its resources. There are a large number of exploitable machines on the internet, which have weak security measures, for attackers to launch DDoS attacks, so that such attacks can be executed by an attacker with limited resources against the large, sophisticated sites. The attackers in DDoS attacks always modify the source addresses in the attack packets to hide their identity, and making it difficult to distinguish such packets from those sent by legitimate users. This idea, called IP address spoofing has been used in major DDoS attacks in the recent past.

2. IP Spoofing overview

The basic protocol for sending data over the Internet network and many other computer networks is the Internet Protocol ("IP"). IP spoofing or Internet protocol address spoofing is the method of creating an Internet protocol packet or IP packet using a fake IP address that is impersonating a legal and legitimate IP address. IP spoofing is a method of attacking a network in order to gain unauthorized access. The attack is based on the fact that Internet communication between distant computers is routinely handled by routers which find the best route by examining the destination address, but generally ignore the origination address. The origination address is only used by the destination machine when it responds back to the source [2].

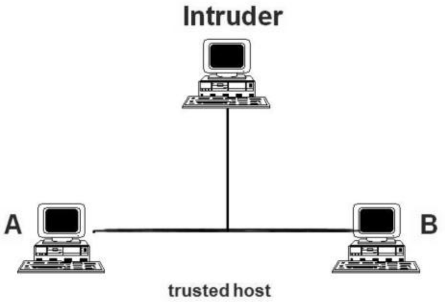

Fig.1. Intruder in communication

In a spoofing attack, the intruder sends messages to a computer indicating that the message has come from a trusted system. To be successful, the intruder must first determine the IP address of a trusted system, and then modify the packet headers to that it appears that the packets are coming from the trusted system. These include obscuring the true source of the attack, implicating another site as the attack origin, pretending to be a trusted host, hijacking or intercepting network traffic, or causing replies to target another system. Spoofing of network traffic can occur at many layers. Examples include network layer spoofing (e.g. Ethernet MAC spoofing), non-IP transport layer spoofing (e.g. IPX, NetBEUI), as well as session and application layer spoofing (e.g. email spoofing). All of these have significant security concerns.

2.1 IP Address Spoofing Attacks

Blind spoofing- This attack may take place from outside where sequence and acknowledgement numbers are unreachable. Attackers usually send several packets to the target machine in order to sample sequence numbers, which is doable in older days. Using the spoofing to interfere with a connection (or creating one), that does not send packets along your cable [3].

Non-Blind spoofing- This type of attack takes place when the attacker is on the same subnet as the victim. The sequence and acknowledgement numbers can be sniffed, eliminating the potential difficulty of calculating them accurately. The biggest threat of spoofing in this instance would be session hijacking. This is accomplished by corrupting the data stream of an established connection, then re-establishing it based on correct sequence and acknowledgement numbers with the attack machine. Using this technique, an attacker could effectively bypass any authentication measures taken place to build the connection [3].

Man in the Middle Attack- This is also called connection hijacking. In these attacks, a malicious party intercepts a legitimate communication between two hosts to controls the flow of communication and to eliminate or alter the information sent by one of the original participants without their knowledge [3].

Denial-Of-Service- To make tracing and stopping the DoS is difficult when the attacker spoof source IP addresses. When multiple compromised hosts are participating in the attack, all sending spoofed traffic; it is very challenging to quickly block the traffic. IP spoofing is almost always used in denial of service attacks (DoS), in which attackers are concerned with consuming bandwidth and resources by flooding the target with as many packets as possible in a short amount of time [3].

3. Related Works

Many approaches against IP spoofing have been proposed by researchers recently. Ingress filtering is a technique used to make sure that incoming packets are actually from the networks that they claim to be from [4].

Egress filtering is the practice of monitoring and potentially restricting the flow of information outbound from one network to another. Typically it is information from a private TCP/IP computer network to the Internet that is controlled [4].

Next approach for filtering spoofed IP packets, called Spoofing Prevention Method (SPM). The method enables routers closer to the destination of a packet to verify the authenticity of the source address of the packet. This stands in contrast to standard ingress filtering which is effective mostly at routers next to the source and is ineffective otherwise. In the proposed method a unique temporal key is associated with each ordered pair of source destination networks (AS's, autonomous systems). Each packet leaving a source network *S* is tagged with the key *K*(*S;D*), associated with (*S;D*), where *D* is the destination network. Upon arrival at the destination network the key is verified and removed. Thus the method verifies the authenticity of packets carrying the address *s* which belongs to network *S*. An efficient implementation of the method, ensuring not to overload the routers, is presented [5]. The major benefits of the method are the strong incentive it provides to network operators to implement it, and the fact that the method lends itself to stepwise deployment, since it benefits networks deploying the method even if it is implemented only on parts of the Internet. These two properties, not shared by alternative approaches, make it an attractive and viable solution to the packet spoofing problem.

SAVE (Source Address Validity Enforcement) protocol when employed enforces all IP packets to carry correct source address. Source Address Validity Enforcement protocol (SAVE) is based on the building an incoming table that consists of association of each incoming interface of the router with different valid source address block. If such tables are deployed at many routers, choices of spoofing addresses reduced to great extent. Every router has a forwarding table that indicates the outgoing interface for a given destination. SAVE suggests that there must be an incoming interface for a source address. Suggesting all packets from specified address space can be reach to destination indicated in incoming table of the router [6].

In Hop-count filtering, an attacker can forge any field in the IP header, he cannot falsify the number of hops an IP packet takes to reach its destination. An Internet server can easily infer the hop-count information from the Time-to-Live (TTL) field of the IP header. Using a mapping between IP addresses and their hop-counts, the server can distinguish spoofed IP packets from legitimate ones. Based on this observation, we present a novel filtering technique, called *Hop-Count Filtering* (HCF)—which builds an accurate IP-to-hop-count (IP2HC) mapping table—to detect and discard spoofed IP packets. HCF is easy to deploy, as it does not require any support from the underlying network [7].

In Updated Hop Count Filtering, the victim can detect and discard the spoofed packets and forward the information to each neighbor routers. It is updated version of hop count filtering.

The probabilistic packet marking (PPM) algorithm was originally suggested by Burch and Cheswick [8] and was carefully designed and implemented by Savage *et al.* [1] to solve the IP trace back problem. It is a used to discover the Internet map or an attack graph during a distributed denial-of-service attack. The PPM algorithm consists of two procedures: The packet marking procedure and graph reconstruct procedure. In the packet marking procedure the packets randomly encode every edge of the attack graph and the graph reconstruction procedure obtains the constructed graph from this encoded information. Here the constructed graph should be the same as the attack graph. The constructed graph is the graph obtained by the PPM algorithm and attack graph is the set of paths the attack packets has been traversed.

In the packet marking scheme the "identification" field of an IP packet is modified which is 16 bits in length. A router marks last "n" bits of its IP address in the IP identification field of the packet it forwards in a "n" bit marking scheme. The identification filed is divided into $16/n$ sections. For indexing section of the field mark, value of packets TTL modulo $16/n$ is used. On receiving packet on one of its interface a router insert marking into identification field using TTL value of t he packet as an index. In case of attack the victim can filter packets based on Pi markings. Vitim has to classify a single packet as an attack packet; victim then records the marking from same packet and further drops all packets carrying same marking [9].

In the marking based detection and filtering scheme, A router puts its IP address into the marking space of each packet it receives; if there is already a number in that space, it calculates the exclusive-or (XOR) of its address with the previous value in the marking space and puts the new value back. This method ensures that the marking does not change its length when a packet travels over the Internet, so the packet size remains constant. To make the marking scheme more effective, let each router perform a Cyclic Shift Left(CSL) operation on the old marking Mold and compute the new marking as M = CSL(Mold)_MR. In this way, the order of routers influences the final marking on a packet received by the firewall. when a packet arrives at its destination, its marking depends only on the path it has traversed. If the source IP address of a packet is spoofed, this packet must have a marking that is different from that of a genuine packet coming from the same address. The spoofed packets can thus be easily identified and dropped by the filter, while the legitimate packets containing the correct markings are accepted [10].

4. Proposed Work

There have been numerous techniques for filtering the spoofed packets and tracing the attack source. But the pure host based mechanism cannot trace the attack source. To trace the attack source, the host based mechanisms have to combine with router based mechanisms. The proposed scheme combines the two techniques, Updated Hop Count Filter and Efficient Probabilistic Packet Marking Algorithm.

Updated Hop Count Filter

The Updated hop count filtering (UHCF) mechanism is used to identify the spoofed packet out of numerous legitimate packets. Whenever source wants to assess the authenticity of any packets then it initiates the verification modules. Initially source wants to communicate with the destinations node then it checks its routing table. It the entry is found then TTL field is updated in initial message. If the entry is not found then it as sends the Multicast Probe RREQ message to destination. Destinations reply with its IP Address, mapping and required details in Probe RREP message. This entry of multicast route is getting updated in routing table. Total number of hops is the number of devices traversed during this data communications. A timer counter is attached with probe message so as to get the validity on time which verifies the route existence. Each device reduces the TTL value by 1 when a packet is transferred from it to any other device. Now the hop count table is created at source end. Now the filtering is applied according to which hop count is calculated as current measured TTL value is subtracted from initial TTL value. Here Initial TTL value is taken from the OS service port number which is fixed. Now the filter selects the TTL value from the table which is just above the measured value.

Hop Distance to Source Node= 255 (Default Initial Value)-Current TTL Value

The hop count of received packet is calculated as t0-t. After the hop count is calculated then the path is checked by condition:

> *Check Path Length (TTL of Stored Hop Count Calculated by Probe Message- TTL of Measured Hop Count by Current Message) = Variable Threshold Value (0 to Number of Multicast Path) &&<=30;*

This condition is verifying the TTL value in which if the differentiated value is lesser than 30 than it is a legitimate route. But in some cases route can of more hops than an average variable threshold is also calculated which lies in between each hops of multicast path. So if the multicast reply came then this condition gets activated which should be above a threshold. From this multipath solution to larger hops is also feasible form up[dated HCF mechanism. Now if the above condition is found to be correct than the packet is taken as a legitimate packet of else it is a spoofed packet. This information is then forwarded to each neighbor so that routing table and HCF value is updated at each nodes and devices.

Algorithm

(i) Send Multicast Probe Message
(ii) Reply Multicast Probe Message (Route Hop Counts 1, Route Hop Count 2,…..Route Hop Counts 3)
(iii) Create Hop Count Table at Hosts (IP Address, Hop Counts, and Low level Interrupts timers)
(iv) Probe message reply comes in a Time Limit (Path Exist) Else Invalid Path
(v) Apply Hop Count Filtering (Checks Spoofed Packet or Not)
(vi) Hop Count = Initial TTL value - Final TTL value
(vii) Checks Hop Count Based on Ports Service
(viii) Select the Port Number Having respective TTL Minimum Above Larger Value from the Current TTL
(ix) The hop count can be calculated for the received packet as follows: (*hop count*) = $t0 - t$. For example, when a host

receives a packet with a TTL value of 120 (*t* = 120), the minimum number in Table 1 that is larger than *t* is 128 (*t0* = 128). Therefore, the hop count is 8 (128 − 120 = 8).
(x) Hop Distance to Source Node= 255 (Default Initial Value)-Current TTL Value
(xi) Check Path Length (TTL of Stored Hop Count Calculated by Probe Message- TTL of Measured Hop Count by Current Message) = Variable Threshold Value(0 to Number of
Multicast Path) && <=30; the packet is legitimate;
(xii) Else
(xiii) Packet is spoofed;
(xiv) Inform Other By Update Alarm Message (Attack Confirm)
(xv) If the difference <30 Packet is legitimate or else Spoofed
(xvi) Inform Other By Update Alarm Message (Attack Confirm)

The detection rate of UHCF consistently swings around the optimum value of 99% which is a good sign of packet filtering technique. So the proposed scheme has chosen this technique to filter the spoofed packet.

Efficient Probabilistic Packet Marking Algorithm

The efficient probabilistic packet marking (PPM) algorithm is used to discover an attack graph during a distributed denial-of-service attack. The EPPM algorithm consists of two procedures: The packet marking procedure and graph reconstruct procedure. In the packet marking procedure the packets randomly encode every edge of the attack graph and the graph reconstruction procedure obtains the constructed graph from this encoded information. Here the constructed graph should be the same as the attack graph. The constructed graph is the graph obtained by the EPPM algorithm and attack graph is the set of paths the attack packets has been traversed.

The router determines how the packet can be processed depending on the random number generated. If x is smaller than the predefined marking probability pm, the router chooses to start encoding an edge. The router sets the start field of the incoming packet to the routers address and resets the distance field to zero. If x is greater than pm, the router chooses to end encoding an edge by setting the router's address in the end field. We use an extra field named as flag which takes either 0 or 1. The flag value at first is made 0 and if the end field is set then the flag is made 1. Now, the start field is encoded only when the flag is 0. If the flag is 1 it implies that the start and end fields together encoded an edge of the attack graph.

Marking procedure at router R

```
for (each packet w received by the router)
{
generate a random number x between [0..1);
if (x < pm and flag=0 ) then
/* router starts marking. flag 0 implies that the packet is not encoded previously */
write router's address into w.start and 0 into w.distance
else
{
If ( w.distance = 0 ) then
write router address into w.end and 1 into flag
}
/* flag 1 implies that the packet has encoded an edge and no other successive routers should
start encoding */
If (flag = 1) then
Increment w.distance by 1
/* w.distance represents the distance of the encoded edge from the victim V */
}
}
```

A victim V, upon receiving packets, first needs filtering of unmarked packets (since they don't carry any information in the attack graph construction). The victim needs to execute the graph construction algorithm for all the collected marked packets and re-construct the attack graph.

Attack Graph Construction Procedure at victim V

```
let G be a tree with root being victim V ;
let edges in G be tuples(start,end,distance);
for (each received marked packet w)
{
if (w.distance==0) then
insert edge (w.start,V ,0) into G ;
else
insert edge (w.start, w.end, w.distance) into G ;
}
remove any edge (x,y,d) with d _ distance from x to V in G ;
extract path (Ri…Rj) by enumerating acyclic paths in G ;
```

A good attack traceback scheme is providing accurate information about routers near the attack source rather than those near the victim. Avoiding the use of large amount of attack packets to construct the attack path or attack tree and low processing and storage overhead at intermediate routers. For these reason, the proposed method has chosen the EPPM algorithm to trace the attack source and intimate to the neighbor routers to prevent further attacks.

So, the proposed method combines the two above techniques for effectively detect, filter the spoofed pocket and also trace the attack source.

5. Conclusion

The number of Internet users is increasing day by day and in the same time the threats in the Internet is also increasing. So security is very important to protect the data and systems from attackers. DDoS attack is one of the dangerous attacks. In recent years various techniques have been proposed for preventing data from DDoS. The Packet filtering mechanisms are only detect and filter the attacked packet, not trace the attacker. The Packet tracing mechanisms are only detect, block and trace the attacked path, not filter. In the proposed method, the victim can effectively detect, filter and also tracing the attacker. In future, we implement this combined approach in MANET.

REFERENCES

[1] S. Savage, D. Wetherall, A. Karlin, and T.Anderson, Practical network support for IP traceback, in *Proceeding of ACM SIGCOMM'00*, Vol.30, No.4, 2000, pp. 295-306.

[2] Yogesh Singh, Hariom Awasthi, Controlling IP Spoofing Through Packet Filtering Using Simulations In Blowfish Algorithm *IJPAERCSE Vol. 01*, Issue 01, 04, 2014.

[3] Mrs. Mridu Sahu Rainey C. Lal ,Controlling Ip Spoofing Through Packet Filtering, *International Journal of Computer Techology & Applications, Vol 3 (1)* 2012.

[4] www.wikipedia.com.

[5] A.Bermlerand H.Levy, Spoofing Prevention Method Proc.IEEE *INFOCOM'05*, 2005.

[6] J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang, Save: Source Address Validity Enforcement Protocol, *Proc. IEEE INFOCOM*, June 2002.

[7] Cheng Jin, Haining Wang and Kang G.Shin, Hop-Count Filtering: An Effective Defense Against Spoofed Traffic, www.eecs.umich.edu/techreporta/cse/2003/CSE-TR-473-3.pdf

[8] H.Rurch and B.Cheswick, *Tracing anonymous packets to their approximate source*, in Usenix LISA, 2000.

[9] Abraham Yaar, Adrian Perrig, Dawn Song, Pi: A Path Identification Mechanism to Defend against DDoS Attacks, *Proceedings of the 2003 IEEE Symposium on Security and Privacy (SP.03).*

[10] Y.Chen, "A *Novel Marking-based Detection and Filtering Scheme Against Distributed Denial of Service Attack*", Masters Paper, University of Ottawa, [1] 2006.

[11] Y. Bhavani, P.Niranjan Reddy, An Efficient Ip Traceback Through Packet Marking Algorithm, *IJNSA, Vol.2*, No.3, July 2010

[12] Mr.Govind M Poddar, Mr.Nitesh Rastogi, UHCF: Updated Hop Count Filter Using TTL Probing and Varying Threshold for Spoofed Packet Separation, *IJERMT, Vol-3*, Issue-4, April 2014.

[13] Mr.Govind M Poddar, Mr.Nitesh Rastogi, Performance Evaluation of UHCF Using TTL Probing for Packet Spoofing Detection in MANET, *IJAREEIE, Vol-3*, Issue-8 August- 2014.