AN ENHANCED IP TRACEBACK MECHANISM FOR TRACKING THE ATTACK SOURCE USING PACKET MARKING AND PATH RECONSTRUCTION

Pavithra K.C1, Dr. Nagesh H.R2

1PG Student, Dept. of CSE, MITE, Moodabidri, India
2Head, Dept. of CSE, MITE, Moodabidri, India

ABSTRACT

The Distributed Denial of Service (DDoS) attacks are launched synchronously from multiple locations and they are extremely harder to detect and stop. Identifying the true origin of the attacker along with the necessary preventive measures helps in blocking further occurrences these types of attacks. The issue of tracing the source of the attack deals with the problem of IP traceback. Several techniques have been proposed for IP Traceback. One of the major challenges in design of efficient traceback scheme is to minimize the number of packets required for successful traceback. Several traceback mechanisms including packet marking schemes have been proposed for DoS/DDoS defense but they have got their own limitations. The main aim of this paper is to reduce the number of packets involved across the efficient traceback by improving the performance of Linear and Remainder Packet Marking schemes using an Enhanced IP Traceback Mechanism (EITM).

Keywords: DoS, DDoS, EITM, LPM, RPM

1. INTRODUCTION

The advances in information technologies in the internet are increasing the possibility of attacks exponentially. There exists several methods towards identifying these attacks, Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are increasing rapidly across the internet world. Denial of Service (DoS) is an attack on availability of a service. The attack aims at denying of an approved service to a legitimate user. When a group of attackers perform DoS attack on a common target, the attack is known as DDoS. The most common method of performing a DoS or a DDoS attacks is to flood the target or network with unwanted traffic, causing interruptions to the communication of legitimate users. DoS/DDoS attacks are posing increasing threat to today’s e-
world. The attacks are evolving in a way, the frequency and the severity, sophistication of DDoS attacks are increasing very fast.

There have been a lot of reports and incidents against these attacks on web based popular applications like Amazon, Yahoo, e-bay, Google and their root Domain Name Servers as well. They include politically and economically motivated attacks on companies causing considerable financial losses and several days of downtime. Initially the internet was designed to provide a communication service; not to handle attacks. The impact of attacks has also increased proportionally with the addition of several high end applications and with the increased use of internet in later years [2]. The task of tracing the attacker is almost impossible because of the stateless nature of internet and destination based type of routing along with IP spoofing [3]. Tracing back the attackers to their source is a task which can make the attackers responsible and come under the scrutiny of law. IP traceback is the mechanism by which victim can trace the attacker to its source and can take preventive measures against the attacker. Identifying and tracing the attacker is considered as an important issue. The source is taken under the security polices of the network and suspected nodes are blocked permanently. IP traceback is a suitable and successful implementation in this regard, where the affected destination node can easily identify the attacker and thus employ required measures to avoid similar attack [4].

Since DDoS attacks are highly distributed in nature, victim can be overwhelmed quite easily even if individual attacker nodes in a group send less number of packets to the victim. The net sums of packets coming from all the attackers that are involved in attack overwhelm the victim and saturate its computing resources. Hence it is seen that in DDoS attacks, individual attackers do not need to send very large amount of packets each. The overwhelming of the victim is accomplished by support from each other. This is the main reason and necessity for a traceback scheme which requires minimal number of packets for traceback. Several IP traceback schemes were introduced and most of them require hundreds/thousands of packets to traceback the attackers [5]. This requirement greatly reduces the effectiveness of IP traceback scheme in preventing and mitigating the DDoS attacks. The success of an IP traceback mechanism mainly depends on the total number of packets required for the IP traceback of attacker nodes and effective mitigation of DDoS attacks [6].

The main aim of this project is to reduce the number of packets involved across the efficient traceback using a mechanism called EITM which requires number of packets almost equal to number of hops traversed by the packet. And the algorithm works efficiently in case if some of the packets in the sequence are lost because of network congestion or any such reason. This framework can be in fact incorporated by other traceback algorithms too to reduce the number of packets required for path reconstruction which can improve their performance as well.

2. LITERATURE SURVEY

IP traceback is used to construct the path travelled by IP packets from source to destination. The source can be a legitimate user or sometimes an attacker. IP Traceback is important and has got various applications. Researchers are prompted to work on IP traceback because of its significance. In this section we review some of the major works done in this field. Burch and Cheswich [7] introduced the concept of IP Traceback for the first time. According to the concept, the attack path was identified by recursively sending packets over the links and monitoring the changes that are seen in the attack traffic. Although this method is simple, it has a few limitations. This method worked only for active attacks. Active attacks are easier to identify. Second point is that traceback itself could initiate a DoS attack [8] as it involved flooding the network with lot of packets. The Fragment Marking scheme [5] proposed by Savage introduced the concept of probabilistic packet marking. But this method required thousands of packets and also intensive computation. Another important scheme proposed by Bellovin is well known as iTrace [9]. iTrace allowed all the intermediate routers in the path send a message to the source/destination of the packet indicating the IP address of...
the router probabilistically. This scheme involved network overhead as extra traffic was generated for traceback. Moreover it required thousands of packets for reconstructing the path. But the good thing about this approach is that packets in flight were not altered. Algebraic traceback [10] suggested by Dean which encoded router’s IP address as a polynomial in IP ID field. But this algorithm does not scale well when there is a DDoS attack. SPIE could perform traceback using just a single packet which was proposed by Snoren [11], but it requires large amount of memory space and also changes to the underlying hardware for packet logging. This was the main reason due to which it could not be implemented in practical.

2.1 An IP Traceback System

Fig 1 depicts network architecture, which might be similar to real time architecture. Consider an example scenario consisting of client 3 sending a message within a data packet to client 10, so the data might travel from client 3 to client 10 through various routers. There are several paths through which the packets can travel. Fig. 1.1 shows the possible paths for the packets to travel from source to destination


Using IP traceback methods the destination node can find the paths travelled by the packets.

![Fig.1: IP Traceback System](image)

IP traceback techniques are basically classified into four different categories. They include Packet Marking, Link Testing, Logging and Messaging. Packet marking technique stores a node’s identification information in the IP header of a packet. This data helps to trace the source of the packet or the path followed by the packet. Existing fields in an IP header are utilized in order to store the marking information; hence, there is no need of changing the structure of an IP header. An IP header has limited space in order to accommodate the traceback data. The data has to be fragmented to overcome this problem. There are different packet marking techniques proposed, which utilize the IP header field to store the traceback data efficiently and help to find the source of the data or the path the packets have traversed.

2.2 Linear Packet Marking

In linear packet marking we use the TTL value of the packet to find the hop count at the router. We use the IP ID field value of packet and hop count of the packet to decide if the router would mark the packet. Using this scheme all the routers get chance to mark the packet in at most 31 packets. Here packets are marked by routers based on values of ID field and hop count of packet at routers in the path. From RFC 791 [12] we see that IP ID field is assigned values normally in three main ways.
1) Sequential- Each session on machine has its own IP ID Assignment counter.
2) Sequential Jump - All the communication sessions going on the machine has same counter for IP ID field assignment.
3) Random - IP ID field is assigned randomly.

Usually due to various reasons like IP header compression, in most of the cases ID field is assigned sequentially or with sequential jump so that differential encoding can be utilized efficiently. Random IP ID field assignment is not that frequent.

2.2.1 Limitations of LPM

LPM performs well in terms of number of packets used for traceback. However it is susceptible to two subtle attacks. They are as follows

1) Constant ID field value:
   ID field value is kept constant for a given session. In that case only one of the routers will mark the packet all the time. For example if the attacker puts ID = 96 again 96% 31 = 3 and only router at hop distance three would mark the packet. So this way he can bypass LPM.
2) If routers chosen at hop count not present in path:
   In this case attacker can find out the distance d in hops to the victim and then assign ID values such that ((ID%31)+1) > d. Hence we will get hops for which routers are not present.

2.3 Remainder Packet Marking

A randomized version of LPM called Remainder Packet Marking (RPM) was proposed by Sourabh and Sairam[1] in order to increase the robustness of the LPM algorithm. Remainder Packet Marking is more robust against possible attacks on LPM. In RPM instead of using a single number 31 for finding which router to mark, we choose k randomly and uniformly from set S = {1,2,3,4,...,31}. This randomization successfully prevents both kinds of attack mentioned above.

1) Constant ID attack: Even if the attacker puts constant value in the ID field, now as our k is randomly selected from 1 to 31, hence all the routers will get a chance to mark the packet.
2) Routers chosen at hop count not present in path: Here again the attacker cannot now devise any such ID value because now k itself will be changing and it will allow all the routers to get a chance to mark the packet.

RPM seems more robust than LPM. But the downside of RPM is that marking field of a packet can be overwritten. In RPM multiple routers can mark the packet because k is chosen randomly at each router and hence the equation (ID%k) +1=hop count can be satisfied at multiple routers. Hence the last router which marks the packet wins. Overwriting a single packet multiple times is not a good idea as it includes extra overhead at the routers. Another drawback of RPM is that it does not take care of packets which are dropped because of some reason. More precisely, RPM does not perform well when a few packets are lost in the sequence.

3. THE PROPOSED ALGORITHM (EITM)

Most of the packets marking techniques available today require huge number of packets for successful path reconstruction. Probabilistic packet marking is used for complete traceback. For edge sampling PPM, expected number of packets X is given by [5]

\[ E(X) \leq \frac{ln(d)}{p(1-p)^d-1} \]  

(1)
where \( d \) is number of hops and \( p \) is the probability of marking by each router and assuming that marks can be overwritten by subsequent routers. If \( p = 1/10 \) and \( d = 10 \) then to reconstruct the path we require 75 packets. We usually overload the 16 bit IP ID field to mark the packet. Usually one packet is not sufficient to convey the entire message. Hence routers send marks in fragments. If each router requires \( k \) fragments then total of \( kd \) fragments are required and expected value of total number of packets required becomes

\[
E(X) \leq \frac{k \cdot \ln(kd)}{p(1-p)^{d-1}}
\]

For example if there are 8 fragments per edge-id, an attack of path length 10 hops and \( p = 1/25 \) requires 1300 packets on average. For node sampling this number is even higher. For \( d = 15 \) and \( p = 0.51 \) we require 42000 packets on average to collect samples from all nodes. Hence we see that we require huge number of packets to perform IP traceback. In DDoS attacks, even though the net sum of attack packets from multiple sources becomes overwhelming for the victim, however to do individual traceback to attack sources we only have fewer number of packets from each source. Hence a technique that does traceback using minimal number of packets which is almost equal to the number of hops travelled by the packet promises to be a very good framework to perform traceback.

EITM works better than LPM and RPM when the source generates sequential ID values. EITM takes care that a packet is marked by only a single router. And every router gets a chance to mark among a maximum of 31 routers. For EITM to work efficiently every marking router first needs to know the total number of intermediate routers between the concerned source and the destination of the packet. So it is essential to get the hop count at each marking router.

### 3.1 Hop Count

The hop count refers to the intermediate devices (like routers) through which data must pass between source and destination, rather than flowing directly over a single wire. Each router along the data path constitutes a hop, as the data is moved from one Layer 3 network to another. Hop count is a rough measure of distance between two hosts. By itself, this metric is, however, not useful for determining the optimum network path, as it does not take into consideration the speed, load, reliability, or latency of any particular hop, but merely the total count.

### 3.2 Path Discovery with traceroute

The program traceroute is a tool used to discover the links along a path. Once packets leave our network, we have almost no control over the path they actually take to their destination. Traceroute is the tool of choice for collecting information related to a path. It is based on a clever use of the Time-To-Live (TTL) field in the IP packet's header. The TTL field is used to limit the life of a packet. The TTL field prevents packets from remaining on a network indefinitely should such a routing loop occur. A packet's TTL field is decremented each time the packet crosses a router on its way through a network. When its value reaches 0, the packet is discarded rather than forwarded. When discarded, an ICMP TIME_EXCEEDED message is sent back to the packet's source to inform the source that the packet was discarded. By manipulating the TTL field of the original packet, the program traceroute uses information from these ICMP messages to discover paths through a network. The program traceroute sends a series of UDP packets with the destination address of the device you want a path to. It sets the TTL field in the first three packets to a value of 1 so that they are discarded by the first router on the path. When the ICMP TIME_EXCEEDED messages are returned by that router, traceroute records the source IP address of these ICMP messages. This is the IP address of the first hop on the route to the destination.

Next, three packets are sent with their TTL field set to 2. These will be discarded by the second router on the path. The ICMP messages returned by this router reveal the IP address of the...
second router on the path. The program proceeds in this manner until a set of packets finally has a TTL value large enough so that the packets reach their destination. Thus the total number of router addresses recorded in a path is denoted as total_hops which is needed for the EITM Algorithm.

3.3 EITM Algorithm

The EITM algorithm is executed by every router on the path from source to destination. And every router checks if it has to mark the packet based on the condition specified in the algorithm. If the condition turns out true then the router inserts its IP address in the marking field (mark_start) and initializes the distance field MHC in the packet to zero. This distance value (MHC) is simply incremented by all remaining routers in the path. This distance indicates the distance of the destination from the marking router. The immediate router which is next to mark_start inserts its IP address in to the field mark_end. And it checks for a MHC value of 0 to do so.

The functionality of a router implementing packet marking technique is shown in Fig 2. Every router in the network executes this packet marking algorithm upon receiving the packet and marks the packet or passes it, based on the output of the algorithm. The receiver might require only packets with unique markings. This might be equal to the total number of hops between the source and the destination.

![Fig. 2: Functionality of a router implementing packet marking](image)

The framework for EITM algorithm has been adapted from the paper [1]. It uses the Identification field of the packet header in an IPv4 datagram to store the identification data of the routers. The stored information is known as marked data. The length of the Identification field is 16 bits. It is split into two parts to store the marking node’s identification data (MRK) and the Marked Hop Count (MHC). MHC is Marked Hop Count indicating the distance from the marking router to the destination. Time to Live (TTL) value depicts the life of the packet on a network in terms of hops. It is known that every router decrements the TTL value of the packet before forwarding the packet to the next router. Initially the MHC is set equal to the Time to Live (TTL) value of the packet by the source. Marked node (MRK) field is used to store the information of the marking router. In order to understand the behavior of the fields MHC, MRK and TTL, consider a scenario of a source, sending a packet to a receiver which passes through intermediate routers Router 1, Router 2 and Router 3 as shown in Fig 3. Let Router 2 be the router which marks the packet.

![Fig. 3: Behavior of the fields MHC, MRK and TTL](image)

- The values of the fields MRK and MHC of a packet do not change until the packet is marked and the value of TTL decrements for every hop.
- The Router 2 that marks the packet by inserting its identification data into MRK field sets the value of MHC to 0 and the value of TTL decrements as usual for every hop.
After the packet is marked by Router 2, the value of MHC is incremented by 1 unit and the value of TTL is decremented by 1 unit for every hop, while the value of MRK is not altered.

From the Fig.3, the receiver can identify that the router with identification value 2 (MRK = 2) has marked the packet is 2 hops away (MHC = 2). If the packet is not marked by Router 2 the values of MHC and MRK remain constant all the way.

An intermediate router upon receiving the packet, checks if the packet has been marked or not. If the packet has not been marked then the router calculates the no of intermediate hops (total_hops) using traceroute. The packet ID is a unique ID assigned to the packet by the source. A modulus operation is performed on the packet ID with ‘total_hops’. Hop count of the packet at the current router is obtained, which is the difference between the Marked Hop Count (MHC) and Time to Live (TTL). If the hop count is equal to the result of (the modulus operation) + 1, then the packet is marked. Here the value of modulus operation is incremented by 1 as the hop count can never be equal to zero. If the hop count is not equal to the result of (the modulus operation) + 1, then the router forwards the packet to the next router. RPM allowed packets to be marked multiple times whereas EITM eliminated this overhead and it is efficient.

4. PATH RECONSTRUCTION USING EITM

In this section we describe the path reconstruction procedure which is more effective than the one used for RPM. In RPM path reconstruction Algorithm shown below, the victim collects attack packets and then constructs a tree using the packet mark information. If the MHC of packet is zero then an edge is added with root as victim otherwise edges in tree are inserted with end point of edge being start and end node given in packet mark and it is inserted at distance P, MHC. We show it in Fig 8. It depicts a victim at root node and three assumed attackers one, two and three. Now attacker 3 is node 8 and its reconstructed path is (1 - 2; 0); (2 -4; 1); (4 -8; 2) where edge is denoted by (start node -end node; distance from root).
But the algorithm does not work out well when some packets in the sequence are lost. For example let us assume there are 5 intermediate routers in the path from A to B. And according to the RPM algorithm B requires a minimum of first 5 packets with sequential IDs for tracing back. We can imagine a case where the 3rd and 4th packets are lost. Now B will not be able to reconstruct the path with remaining set of received packets. EITM solves this problem efficiently. The Path Reconstruction Algorithm of EITM initially checks if there is a packet loss based on the ID values. If so it drops all received packets and waits for the next 5 packets to be received in order to reconstruct the path.

![Modified Path Reconstruction Algorithm](image)

![Path Reconstruction in EITM](image)
5. RESULTS OF SIMULATION

In this section, we try to analyze how RPM and EITM perform with the present PPM Schemes. We simulate our algorithm in MATLAB. We have written small code to simulate PPM, RPM and EITM. The results are given below. Fig.8 shows EITM performs traceback more efficiently in terms of number of packets required for traceback. But when there is packet loss EITM requires more number of packets for traceback.

![Fig.8: Packets required vs. Hop count](image)

PPM is based on the concept of routers probabilistically marking the packets. In most basic PPM called node sampling, each router with probability $p$ marks the packet and the mark can be overwritten by subsequent routers in the path. This becomes a geometric series and probability of marking by router decrease as the hop distance from the source or attacker increases. The probability as function of number of hops is shown in figure 4 given below. We see from Fig.9 that as distance in number of hops from attacker increases, the probability of packet being marked by routers increase geometrically.

![Fig.9: Probability of marking vs. Hops](image)

6. CONCLUSION

The rise of Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks on the internet are growing rapidly causing troubles to the network and also decreasing the network performance. IP traceback is a method to traceback to the source of the packets. Packet marking schemes are the most successful implementation towards preventing DoS attacks by tracing to the source of attacks. Most of the existing packet marking techniques requires huge number of packets to
traceback the source. They also have drawbacks such as router overhead, packet header overload, network overhead, etc. PPM schemes are very promising and perform well in traceback; however they require thousands of packets for performing traceback.

Deterministic packet marking approach can only help in tracing back the nearest edge router which is forwarding the attack packets. We propose a new packet marking based traceback scheme called Enhanced IP Traceback Mechanism that can perform traceback just by using number of packets equal to the number of hops between attacker and the victim. This scheme improves the performance of LPM and RPM by incorporating traceroute utility to find out the number of intermediate routers between concerned source-destination pairs. Algorithm also takes care of lost packets during path reconstruction. As this scheme requires very few packets hence it can perform very well in case of large scale DDoS attacks and even in low rate DoS attacks.

REFERENCES