

IP Traceback to Prevent Denial of Service Attack

Rohit Jain

Abstract—The rising threat of DoS, and DDoS has made IP Traceback an important problem. Instead of preventing systems, we have to deploy mechanism to actually trace the culprit. In this way the attackers will be discouraged. People are working on mechanisms to defend against DDoS. We need mechanisms which are easy to deploy and which require less management work. In this report, we discuss the current scenario of IP Traceback mechanisms. We make a simple matrix on which we analyze different mechanisms.

I. INTRODUCTION

Internet is a stateless system. In Internet Protocol, source host fills source id by itself. Hence source can fill wrong IP deliberately [3]. This act of putting wrong source id in the packet header is called IP Spoofing. This leads to various attacks including Denial of Service. And the only way to know who actually sent the packet is either by tracing the path of the packet or providing separate medium for each host and checking each packet sent by the host. Of course, later is very costly. This leads us to the problem of IP Traceback.

The first question is "What is Denial of Service attack?". Denial of Service attack is an attack on a resource such that its quality of service is hampered. The attacker consumes the host's (which is giving service) resource, and hence either the host goes down or it starts giving less service to other hosts. An example can be SYN attack. To establish a TCP connection SYN packet is sent by one host to another. When the second host receives the SYN packet it sends SYN-ACK back to the first node. And side by side assigns some of its resources for some time for the to-be-made connection. When SYN-ACK reaches the first host, the connection is said to be established. An attacker may send false request for TCP connection to a host. And hence some resources of the host (let us call it victim) are wasted. A worse situation will be when the attacker:

- sends multiple SYN packets for multiple TCP connections
- spoofs its IP and sends packets with different IPs

In both the cases the victim will assign resources for all the different IPs / SYN packet and hence a large amount of resources will be wasted. And hence others who send true requests for resources will be in a worse-off situation. Similar is the case with UDP or ICMP flooding. Another problem with such situations is that if the attacker gives IP of some other host, that host will be flooded by SYN-ACK packets. And hence that node has to struggle as well.

Distributed Denial of Service attack is even worse. In (D)DoS a single culprit uses multiple nodes to send attack packets to single victim. The situation is worse because now it becomes even more difficult to decide whether the packets

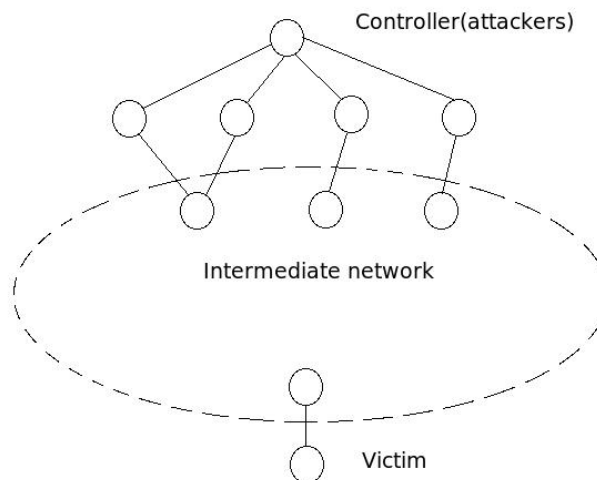


Fig. 1. A typical Distributed Denial of Service Attack in which a single node controls multiple nodes to attack on a victim.

are coming because the attacker is flooding or the packets are true requests. Fig. 1 shows a typical DDoS Model.

To deal with such situations there can be broadly three approaches. One, that we try to eliminate the possibility of the attacks. We do that by either constraining the system to make connections with trusted clients only or by using only secured connections. Second, that we just prevent such attacks [1]. Ways like '*Client puzzles prior to committing resources*' can do that. In this technique the server sends puzzles to the clients when server is lacking resources. If the client is able to solve the puzzle that will mean that client is *pure*. The puzzles must be random so that the attacker may not know the answer before hand. If the client already knows the answer, it can send the solution to the puzzle immediately. This technique ensures that there is no IP spoofing. But it increases the victim overhead. And consumes bandwidth as well. Another way can be that we do *ingress filtering* [2]. This method has been discussed in the Section 4.1.

Third approach can be that we try to find out from where the packets are coming. This way we can catch the defaulter. These techniques not only mitigate denial of service attacks but also try to discourage the defaulters. The problem with first approach is that we put lots of constraints on the system. Second approach has been major field for development. The problem with this approach is that it does not discourage the attacker from attacking. Keeping this in mind, work is being

done on IP Traceback.

IP Traceback is a mechanism in which we trace the path, the packet followed to reach the victim. It is being tried to make mechanisms which could trace the path as close to the attacker as possible. Different techniques have been designed, none implemented in real world (to the best of the author's knowledge). There has been some implementations on simulators [11]

In this report we first discuss (Section II) some areas that needs to be taken care of while designing a mechanism for IP Traceback, i.e. the trade off matrix. Then we discuss some basic assumptions that need to be made while thinking of a mechanism (Section III). Then in Section IV some relevant works have been discussed. Finally in Section V, we discuss some directions in which the developed should go.

II. TRADE OFF MATRIX

A good matrix was designed in [4], in which main emphasis was given on "Management overhead", "Network overhead", "Router overhead", "Preventative/reactive". We extend that matrix to include few more important issues. The main aspects we will be taking into account are :

A. Router Overhead

The only way in which these paths can be traced is involvement of routers. Hence routers work will be increased. But still the overhead should not be much otherwise the processing time will become a big overhead for bandwidth.

B. Packet Size Overhead

Packets may be marked by routers to send information about the path. Hence the packet size can be increased. One basic thing that should be considered is that the size of packet should be static, otherwise fragmentation may occur frequently, which may lead to problems.

C. Network Traffic Overhead

In any case since the information about the path needs to be flowed there will be increase in network traffic. In general this doesn't seem to be a big problem, but for some situations it may become a problem. Mechanisms like "ICMP Traceback" [5] or "Hash-based IP Traceback" [10] may be problematic for some low bandwidth networks.

D. ISP Management Overhead

The main problem to preventing the attacks is that ISPs usually don't implement right systems either because of lack of knowledge or (specially in case of IP Traceback) they don't want to disclose their network arrangements, because disclosing information about their internal networking may lead to security threats. Schemes likes [4] or [8] discloses the internal networking scheme perfectly. So, it is important that the ip traceback mechanism doesn't increase too much of ISP management.

E. Number of packets required to trace the path

Usually the DoS attacks are flooding type attacks. Thousands of packets are sent in the attack. But in some systems, less packets may damage the system as well. For example, Teardrop attack crashes versions of Microsoft Windows with one packet [6]. So number of packets required is an important parameter.

F. Deployment Issues

The major 2 issues related to deployment would be how efficient it is to deploy the mechanism, and secondly if the mechanism supports incremental deployment. Second issue is important, because if incremental deployment is not impossible, it will be practically impossible to use the mechanism in real world.

G. Working Against DDoS

Today DDoS attacks have become frequent. Hence we need to make mechanism to defend not only simple DoS attacks but also DDoS attacks.

All these are major issues that will need to be optimized accordingly. Though there will always be trade offs between them. In this report it has been tried to analyse different mechanisms present today, on these criterions.

III. OVERVIEW

In this survey we will mainly be discussing some marking schemes for IP Traceback. Marking means that hops in the path augments some bits in the packet, which is then used by the victim to trace the path. So basically the system has got 2 components, one is how are routers marking the packet, second is how the victim is reconstructing the path. Number of packets a victim must get to trace the path can be called *convergence time*. Convergence time can be of great importance depending upon the application. We will also be discussing some mechanisms which will make victims overhead less. For doing further discussions we identify some basic assumptions about our model. And then depending upon variation of systems we change the assumptions and discuss algorithms accordingly. Some basic assumptions which we will take and has been discussed in [4] and [7] are :

- An attacker may generate any packet
- Attacker(s) may be aware that they are being traced
- Routers are both CPU and memory limited.
- Routers are not widely compromised

Depending upon the application and requirement we will have to change our constraints. For some systems we will assume that multiple attackers can be there, we will also assume that lots of packets are sent for the attack. The good scheme for ip traceback would be in which *false negatives* are not there and *true positives* are lesser. False negative means that the attacker is declared to be *pure*. And a true positive is that a *pure* host is declared to be attacker/suspect.

IV. DIFFERENT MECHANISMS

A. Ingress filtering

Ingress filtering is not as such an IP traceback mechanism but it deals with defending denial of service attacks. The basic idea is that the ingress router can check if the IP is spoofed or not. Once we are sure that the IP is not spoofed, there is no need to do IP traceback. Routers at the border of ISP networks or Label Switch Router can be assigned this job of legitimating the source address IP of the packet arriving [2].

The system can be very powerful against IP Spoofing and hence DoS if all the ISPs cooperate. Which doesn't seem to be the situation. Most of the ISPs don't support this system. The reason being that the router overhead can be too much. The ingress routers have to have good architecture, so that it doesn't become the bottle neck. And then even if the technology has been implemented an attacker can use someone else's IP which is on the same side of the router (hence the technique will fail), like in the case when several major US web sites (including yahoo and amazon) were attacked in 2000.

B. ICMP Traceback

Since routers generate many ICMP packets, it was suggested [5] that routers should send path information using ICMP packets. The router puts IP of adjacent router on the path and its own IP on the ICMP packet. It also puts the data packet header in the ICMP packet so that the victim could identify for which packet did the ICMP message came. The router also embeds TTL, so that the victim could reconstruct the path. The router sends ICMP with very less frequency. Hence only the flooding type attacks can be prevented. If its done in higher frequency then a lot of bandwidth wastage will be there. Another problem is that in todays systems in case of attacks the system usually discards the ICMP messages hence the information embedded in ICMP message can not be used. Another problem is that (note the first assumption) the attacker may send false ICMP messages. In that case the mechanism will fail. Though some key distribution may prevent the second problem but that will increase management overhead. Since the messages used for traceback are ICMP, we need not care about backward compatibility.

C. Probabilistic Packet Marking

1) *Edge Sampling*: This scheme takes into account (apart from the assumptions made above) that multiple attacker can be there (that means it can be used to defend against DDoS), and it assumes that the attack is flooding type i.e. millions of packets are sent by the attackers. The scheme [4] says that the packet should be marked, with some probability, with an edge of the path and its distance from the victim. Consider the case in fig. 2. A1, A2 ... are attackers, and R1, R2 etc. are the connecting hops. V is the victim. The scheme adds an overhead to the packet. Good thing is that the overhead is constant hence different effects can be predicted beforehand. The IP header needs to be appended with 72 bits (in case of IPv4). 32 bits for each end of the edge (call it ID1 and ID2, so

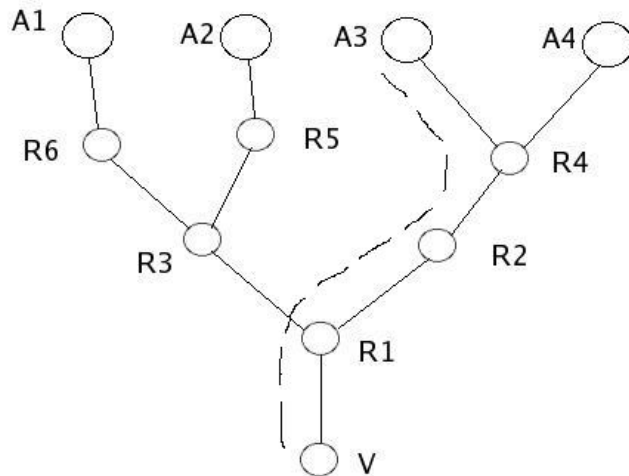


Fig. 2. A case where multiple attackers are there. A1, A2..... are attacker and V is representing victim. The dotted line represents a path that the packet will follow to reach victim.

for the edge R4-R2, ID1 is R4 and ID2 is R2), and 8 bits for distance (for R4-R2 it will be 2). The algorithm for marking is like this: when router gets a packet it marks the packet with some probability, say p . Marking means that the router puts its IP in ID1. Whenever a router finds that the last router marked the packet (i.e. ID2 is 0 and ID1 is nonzero) it puts its IP in ID2 and increases the distance by 1. If a router is doing none of the above and the ID1 is nonzero it increases the distance field by 1. The reconstruction algorithm is easy too, though it is too complex for computation. The mark which is most frequent will be of the nearest router. The frequency of a mark decreases with the distance of the corresponding edge. Using this method path is reconstructed. With enough number of packets the path can be generated easily. But the big distributed denial of service attack may take days to reconstruct the path. And there will be many *false positives*. The main problem with the scheme is that (note the first basic assumption) the attacker may fill these augmented fields and hence can misguide the reconstruction procedure (more false positives). Though at the same time it should be noted that some part of the path can be predicted anyways. With distance field set to 8 bits we restrict that the path can not have more than 256 hops, which makes sense, as in todays internet situation 256 hop count is a big amount. If some part of the network doesn't support this method then as well, the method will work. A major property of this scheme is that router overhead is very less. As the marking can be done on the way, lag wouldn't be there at routers. Another advantage of the scheme is that progressive deployment is not a problem. If suppose R2 doesn't support the scheme, in that case the victim will find R4-R1 as edge and not R2-R1 and R4-R2. And hence the victim will still be able to trace the path to the attacker(s). Now probability that

the packet is marked by the edge at a distance d is

$$\rho(1 - \rho)^{d-1}$$

, and hence using coupon collector problem it can be shown that the expected number of packets required will be less than

$$\frac{\ln(d)}{\rho(1 - \rho)^{d-1}}$$

2) *Advanced and Authenticated Marking Schemes for IP Traceback*: Advanced marking scheme [8] is improvement on [4]. The scheme follows Fragment Marking Scheme [4]. It is assumed that the victim knows the upstream routers map. The assumption is genuine as tools are available that gives the upstream router map, like *traceroute*. Since the map is known to the victim, it doesn't need 32 bits for representing a router uniquely. It uses a hash to convert the 32 bits to 11 bits. Since total number of routers on the map usually wouldn't be too much, we can find a hash function which is a *good* hash function. A good hash function is a function in which number of collisions are less. It uses 2 hash functions, one for upstream router of the edge and other for downstream node of the edge. It helps in avoiding collisions. The scheme augments the packet datagram with just 16 bits as compared to 72 bits in PPM. It uses 11 bits for sampling edge and 5 bits for distance. 5 bits means that there can be at most 32 hops in the path, which is not less for the current internet structure. Now instead of appending ids of both routers, it XORs the ids. Hence just 16 bits are required. 2 important observations are: 1: that $a \oplus b \oplus a = b$. 2: that when the last router in the path marks the packet (i.e the outgress router) the 11 bits are not XORed by any other id. Hence the victim straightaway gets the outgress routers id. Then the edge at a distance 1 would be XOR of outgress router and one before that, so that id can be found out, and so on. In the fig.2 the victim easily see the id of R1 (as soon as it gets a packet with distance 0). Then as soon as it gets packet with distance 1, it can get id of R2. Here there comes a problem when $R2 \oplus R1 = R3 \oplus R1$. Though the paper clearly shows that this wouldn't be a problem as probabilistically it can be shown that if for a map having distance of maximum length path as 32 and maximum number of children for the map be 64 (a very bad case) the expectations for collision will be less than 1. It comes out that instead of thousands of packets required in [4], less than thousand packets are required in this scheme. But the scheme increases reconstruction overhead. Like was the case with PPM, if a part of network doesn't support this mechanism then as well, the technique can be used.

D. Tabu Marking Scheme

Tabu Marking Scheme [9] improved upon advanced marking scheme. The scheme says that if a packet has already been marked it should not be marked. In this way packets reaching victim with the mark of the router at the maximum distance will be more than the packets with mark of the nearer router. The paper proves that the expected number of packets required

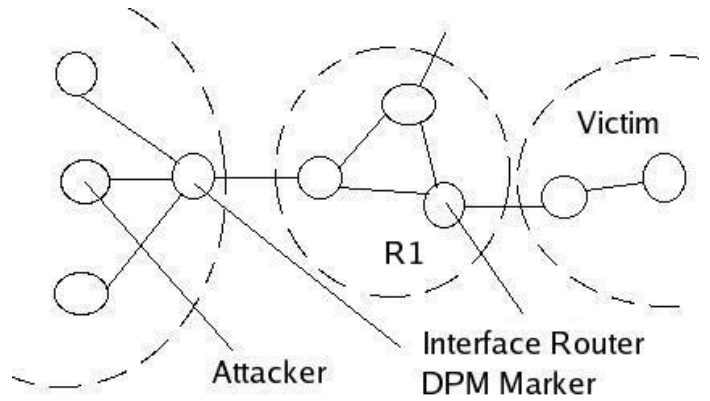


Fig. 3. A typical situation for DPM, where the router which are network interface does marking.

Version	Header Length	Type of Service	Total Length	
Identification			DM FF	Fragment Offset
TTL	Protocol		Checksum	
Source Address				
Destination Address				
Options				
Payload				

Fig. 4. A typical situation for DPM, where the router which are network interface does marking.

to trace the path in case of Tabu scheme is less than in case of Advanced Marking Scheme. Though since the marks can be spoofed (note the first basic assumption) some keys have to be shared between routers, so that the attacker couldn't spoof mark. This leads to different security concerns. Key distribution leads to management overhead as well.

E. Deterministic Packet Marking

Till now the algorithms/schemes discussed here assumed that the path remains the same. But in real world that may not be true. It is argued in [7] that the nodes we can be sure of are the interface routers of the ISPs only, ISPs may have there own internal IPs for internal networks, which wouldn't be of use and the internal path may be changing as well. It presents a different sort of algorithm which actually isn't efficient if we look the way it marks network segments suspected to be attacking. The scheme says that all the egress router should mark the packet. Hence when the packet reaches victim, it is

been marked by the nearest egress router on the path. Consider the fig.3. The mark on the packets sent by the attacker will be of router R1. Now whenever the victim suspects that an attack is being done, it blocks every packet coming from R1. This may surely stop the attack, but at the same time blocks many genuine clients i.e. true positives are too much. Every connection through R1 is broken. The big advantage for this scheme is that the attack can easily be stopped. The paper suggests that the routers should put one half (randomly) of their IP on the packet and a flag which says which half it is. In this way for 99% confidence of being able to trace path, we need just 7 packets. Not much of router overhead is there. Even expected number of packets required to trace the attacker is very less. The scheme ensures that attacker can not spoof mark.

F. Hash-Based IP Traceback

This scheme [10] is different from what we have discussed till now. In this scheme there is no overhead for the victim. All victim needs to do is to ask the Source Path Isolation Engine (SPIE) to trace the packet. SPIE keeps record of every packet on network. Routers send a digest for every packet to SPIE. And the SPIE keeps record of latest digests. A digest is a part of IP packet which doesn't vary from router to router. Gray fields in fig.4 are masked before coming in digest. The digest of a packet contains the header of the packet and first 8 bytes from its payload. The paper shows that these 28 bytes, are sufficient to differentiate 2 packets normally. It shows, experimentally, that fraction of collided packets is merely 10^{-5} . Since it is not possible to store all the digests coming from all routers, the scheme uses *Bloom Filter* to store digests. Bloom filter computes k distinct packet digests for each packet using independent uniform hash functions which give n bits. And these are indexed into a 2^n size 1 bit array. Now at the time of lookup, i.e. when we need to check whether the packet was sent through the router, SPIE generates the n bit number for the packet to be checked and checks if the array bit at this index is 1 or 0. If it is 1 then its most likely that the packet was sent. The scheme is very good from victims point of view, as it has nothing to do for traceback. Though bandwidth overhead is there. Every router has to send one extra packet for every incoming packet. We have to make one SPIE, which increases ISP management overhead. Number of packets required is not high as almost one to one hash functions (good hash function) are used. The major drawback would be in deployment as many nodes have to be assigned the job of SPIE.

V. CONCLUSION

Today we can find many tools for doing DoS attacks. DoS attacks have become very popular. Hence we need to design proper mechanisms to protect systems from such attacks. Mechanisms has been developed and deployed to prevent such attacks. But DDoS is still a problem as it is difficult to trace DDoS attackers and its effect is too bad. We need to start development towards defending DDoS. Some schemes are present which very well defends such attacks, but without the

cooperation of ISPs it will be difficult to deploy any scheme. Though RFC asks to deploy ingress filtering, still very less number of ISPs have deployed that. Mechanisms like hash-based traceback leads to many management issues, which in current scenario doesn't seem to be working. Mechanisms are there which talks about single packet traceback, but there are lots of overheads for such methods.

REFERENCES

- [1] O. Spatscheck and L. Peterson, "Defending against denial-of-service attacks in Scout", in Proc. USENIX/ACM Symp. Operating System Design and Implementation, Feb 1999, pp. 59-72
- [2] P. Ferguson and D. Senie, "Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing", IETF, RFC 2267, Jan 1998.
- [3] R. T. Morris, "A weakness in the 4.2BSD Unix TCP/IP Software", AT & T Bell Labs, Tech. Rep. Comput. Sci. 117, 1985
- [4] S. Savage, D. Wetherall, A. Karlin and T. Anderson, "Network support for IP traceback", IEEE/ACM Transactions on Networking, vol 9, no. 3, pp. 226-237, June 2001.
- [5] Steven M. Bellovin, "ICMP Traceback Messages", Internet Draft, March, 2001.
- [6] Microsoft Corporation. Stop 0A in tcpip.sys when receiving out of band (OOB) data. [Online] Available at : <http://support.microsoft.com/support/kb/articles/Q143/4/78.asp>
- [7] A. Belenky and N. Ansari, "IP traceback with deterministic packet marking", IEEE Communications letters, vol 7, no. 4, pp. 162-164, april 2003
- [8] D. X. Song and A. Perrig, "Advanced and Authenticated Marking schemes for IP Traceback", Proc. INFOCOM, 2001, vol. 2, pp. 878-886.
- [9] Miao Ma, "Tabu Marking Scheme for IP Traceback", 19th IEEE IPDPS, 2005.
- [10] A. Snoeren, C. Patridge, L. Sanchez, C. Jones, F. Tchakountio, S. Kent and W. Strayer, "Hash-based IP Traceback", in Proc. ACM SIGCOMM'01, San Diego, CA, Aug. 2001, pp. 3-14.
- [11] Vrizzlynn L. L. Thing, Henry C. J. Lee, "IP Traceback for Wireless Ad-hoc Networks", VTC, IEEE 2004.

	<i>Router Overhead</i>	<i>Packet Overhead</i>	<i>Network Overhead</i>	<i>ISP Overhead</i>	<i>Packets Required</i>	<i>Progressive Deployment</i>	<i>Against DDoS</i>
Ingress	large for interface router	null	null	management of interface router	null	Not possible	bad, many false positive
ICMP	not much	null	high for high frequency	not much	much	will be difficult	not much
PPM	not much	not much	no	not much	>1000	possible	ok for 20-30
AAMS	Less than PPM	less than PPM	no	not much	1000	possible	perfect
TMS	not much	not much	no	not much	<1000	possible	perfect
DPM	large for interface routers	not much	management overhead	management of interface router	<10	true positives problem	no
SPIE	too much	null	yes, extra packets are sent	yes, SPIE needs to be deployed	much	possible	good

Fig. 5. The table compares different algorithms discussed in this report, on the basis of the matrix we chose initially.