ISSN:0975-9646

An Accurate Termination Condition of the Probabilistic Packet Marking Algorithm

J.Shiva Kumar, Prof P.Pradeep Kumar

Department of Computer Science & Engg, Vivekananda Institute of Tech & Sciences, JNTUH, Karimnagar,Hyderabad, AP, INDIA

ABSTRACT- **Denial-of-service (DoS) attacks pose an increasing threat to today's Internet. One major difficulty to defend against Distributed Denial-of-service attack is that attackers often use fake, or spoofed IP addresses as the IP source address. Probabilistic packet marking algorithm (PPM), allows the victim to trace back the appropriate origin of spoofed IP source address to disguise the true origin. In this paper we propose a technique that efficiently encodes the packets than the Savage probabilistic packet marking algorithm and reconstruction of the attack graph. This enhances the reliability of the probabilistic packet marking algorithm.**

KEYWORDS- **Denial-of-service, Probabilistic Packet Marking Algorithm, Efficient Probabilistic Packet Marking algorithm, attack graph.**

1. INTRODUCTION

Defending against Denial-of service attacks is far from an exact or complete science. Rate limiting, packet filtering [4], [6], [7], and ICMP traceback [3], in some cases, help limit the impact of Denial-of-service attacks, but usually only at points where the Denial-of-service attack is consuming fewer resources than that are available. In many cases, the only defense is a reactive one, where the source or sources of an ongoing attack are identified and prevented from continuing the attack.

One major difficulty is to defend against Distributed Denial-of-service attack is that attackers often use fake, or spoofed IP addresses as the IP source address. Therefore, attackers can easily disguise themselves as some other hosts on the Internet. Because of the stateless nature of the Internet, it is a difficult task to determine or trace the source of these attacker's packets and there by locate the potential locations of these attackers. This is known as the IP traceback problem.

Many IP traceback techniques [8], [10], [11], [12], [14] have been proposed, they all have short comings that limit their usability in practice. Some of them are Ingress filtering[5] requires edge routers to have sufficient processing power, to inspect the packet's destination IP address for normal packet forwarding service. It also need to inspect the source address and determine whether it is a legitimate or illegitimate address. Another major problem with ingress filtering is that this technique is only effective if there is a widespread deployment in the networking community such that many ISPs are willing to deploy this service. Moreover, even with the enabling of ingress filtering service, attackers can still forge the source IP addresses as other hosts within their network domain. Alternative approach to DDoS traceback includes input debugging approach [18] which requires cooperation between system administrators of different ISPs. Therefore, it may not be able to trace the attackers in realtime or in the midst of a DDoS attack. Other approaches such as controlled flooding [16], which either generates many additional packets to the network (which can be viewed as another form of DDoS attack), or network logging [11], which requires additional storage and computational overhead of the participating routers. All, the above approaches have performance problems and significant deployment difficulties.

One promising solution, proposed by savage et al [9], is to let routers probabilistically mark packets with partial path information during packet forwarding. The victim then reconstructs the complete path after receiving a modest number of packets that contain the marking. This approach has a low overhead for routers and the network and supports incremental deployment. We call this type of approach as the IP marking approach. In this paper we propose a new scheme similar to the technique used by Savage. The difference is our technique significantly encodes all the edges needed by the victim to reconstruct the attack path.

This paper is organized as follows. In section 2 we describe about PPM algorithm. Section 3 presents related work. Section 4 introduces the modified packet marking algorithm (EPPM) concrete encoding strategy and implemented with our new algorithm. Section 5 presents experimental results of our work. Finally section 6 describes conclusion and future scope of the work.

2. THE PROBABILISTIC PACKET MARKING ALGORITHM

The probabilistic packet marking (PPM) algorithm was originally suggested by Burch and Cheswick [16] and was carefully designed and implemented by Savage *et al.* [9] to solve the IP traceback 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.

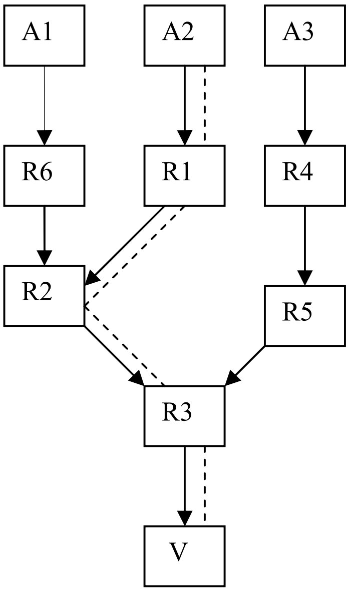


Figure 1. An attack graph containing attack path.

The network can be viewed as a directed graph G = (V,E) where V is the set of nodes and E is the set of edges. V may be a single host under attack, or a network border device such as a firewall or intrusion detection system that represents many such hosts. Every potential *attack origin* Ai is a leaf in a tree rooted at V and every router Ri is an internal node along a path between some Ai and V. The *attack path* from Ai is the ordered list of routers between Ai and V that the attack packet has traversed, e.g. the dotted line in the figure 1 indicate the attack path: (R1, R2,R3). The *distance* of Ri from V on a path is the number of routers between Ri and V on the path, e.g. the distance of R1 to V in the path (R1,R2,R3) is 2. The *attack graph* is the graph composed of the attack path e.g., the attack graph in the example will be the graph containing the attack path (R1,R2,R3). And we refer to the packets used in DDOS attacks as attack packets.

2.1 Packet Marking Procedure

To implement an IP traceback service previously they used to allocate enough space in an IP packet header so that one can use this space to record the traversed path of a packet. For example, each router, beside performing the normal packet forwarding and routing functions, records or appends its own ID in the pre-allocated space at the packet's header. In this analogy when a victim receives a marked packet, victim can examine the packet's header and obtain the complete traverse path information of the marked packet. However, one major problem about this simple approach is that the length of a traversed path (e.g., number of hops) of a packet is not fixed. Therefore, it is impossible to pre-allocate sufficient amount of space in the packet's header in advance. Another technical difficulty of recording complete path information of each packet to the victim is that if an attacker can potentially manipulate this path information and fill in false router's identification in the packet's header it misleads the victim site.

The packet marking algorithm proposed by Savage [9] instead of recording the complete path information of a packet, only records each edge traversed from the attacker to the victim site in a probabilistic fashion. The routers encode the information in three marking fields of an attack packet: (start, end, distance). The start and end fields store the IP addresses of the two routers at the end points of the marked edge. The distance field records the number of hops between the marked edge and the victim site.

In the PPM a packet stores the information of an edge in the IP header. The pseudocode of the procedure [9] is given in Fig. 2 for reference. 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.

```
Marking procedure at router R
for each packet w
let x be a random number from [0..1)
if x < $p_m$ then
write R into w.start and 0 into w.distance
else
if w.distance = 0 then
write R into w.end
increment w.distance
```

Figure 2. packet marking algorithm.

2.2 Graph reconstruction procedure

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. Figure 3 illustrates the attack graph construction algorithm.

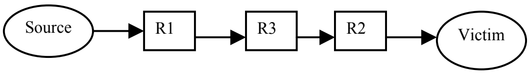
```
Attack Graph Construction Procedure at victim V
let G be a tree with root being victim V ;
let edges in G be tuples(start,end,distance);
for (each received marked packet w)
{
if (w.distance==0) then
insert edge (w.start,V ,0) into G ;
else
insert edge (w.start, w.end, w.distance) into G ;
}
remove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

Figure 3. Attack Graph Construction algorithm.

3. RELATED WORK

In the packet marking procedure, even if a packet has already encoded an edge, successive routers may choose to start encoding another edge randomly. As a result, when a packet arrives at the victim, it may either encode any of the edges of the attack graph, or may not encode any edge.

Figure 4 illustrates the set of marked and unmarked edges collected by the victim under a simple linear network topology. In this example, the victim could collect 4 types of packets.



Marking field	data

Start	End	d	☒

R1	R2	2	☒

R2	R3	1	☒

R3	-----	0	☒

----	-----	----	☒

Figure 4. Example of packet marking procedure

According to the probabilistic packet marking algorithm, each packet may mark or unmark an edge with some probability. Let $P_m(d)$ denote that an edge is marked, and it is d hops away from the victim site. In general, we have

$P_m(d) = p(1-p)^d \quad d \geq 0$ ------------ (1)

In some cases the packet may not be encoded at all, this is the case when in every router x is less than Pm. Let Pu(d) be the probability that a victim V will not find an edge which is d hops away as a marked edge. We have

$P_u(d) = (1-p)^{d+1} \quad d \geq 0$ ---------------(2)

In other words, all routers along the path to the victim decide not to mark the packet. So figure 4 shows the marked packets with marked edges (R1, R2), (R2, R3), and (R3, -). The victim V can also receive unmarked packets.

This paper mainly presents the effective way of encoding the edges. In the Savage[9] algorithm when a packet arrives at a router R1, the router determines how the packet can be processed based on a random number x (line number 1 in the Figure 2). If x is smaller than the predefined marking probability P_m, the router chooses to start encoding an edge. The router sets the start field of the incoming packet to the router's address and resets the distance field of that packet to zero. Then, the router R1 forwards the packet to the next router R2 as shown in the figure 5.

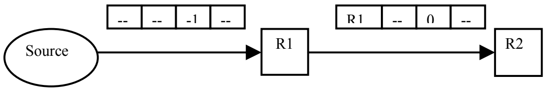


Figure 5. Packet w received to R2.

When the packet arrives at the router R2, the router R2 again chooses if it should start encoding another edge. For example, for this time, let us suppose the router chooses not to start encoding a new edge (it is the case when x is greater than Pm). Then, the router R2 will discover that the previous router R1 has started marking an edge, because the distance field of the packet is zero. Eventually, the router R2 sets the end field of the packet to the router's address as shown in figure 6. Then this router R2 again forwards the encoded packet to the next router R3. Now at R3 if x is smaller than the predefined marking probability P_m again it will start encoding an edge but this shouldn't happen. The process is shown in the figure 6.

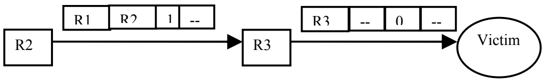


Figure 6. Packet w received to victim

A good traceback scheme should provide accurate information about routers near the attack source rather than those near to the victim. This is the pivotal drawback in the savage [9] algorithm. To overcome this drawback we described a modified algorithm in the next section and named it as an efficient probabilistic packet marking (EPPM) algorithm.

The graph reconstruction procedure is started as soon as the victim starts collecting marked packets. When a marked packet arrives at the victim, the procedure first checks if this packet encodes a new edge. If so the procedure accordingly updates the constructed graph G. From the above example we get some sample packets as shown in figure 7.

R2	R3	1	☒

R1	R2	2	☒

R3	---	0	☒

R2	R3	1	☒

Figure 7. Sample of packets

Then we extract path by enumerating acyclic paths in G and construct the attack graph as shown in the figure 8.

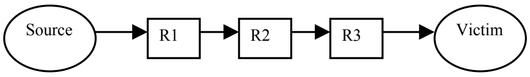


Figure 8. The attack graph

4. PROPOSED ALGORITHM

In our proposed algorithm, as shown in the figure 9 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. The packet traverses from source to R1 to R2 and then to R3 as similar to in the previous section and assume that the encoding is also the same but after the packet received at R3. R3 cannot start encoding again since the flag value is 1. As the successive routers cannot start encoding that packet again, just they increment the distance field so that the victim can know the distance of the encoded edge from it.

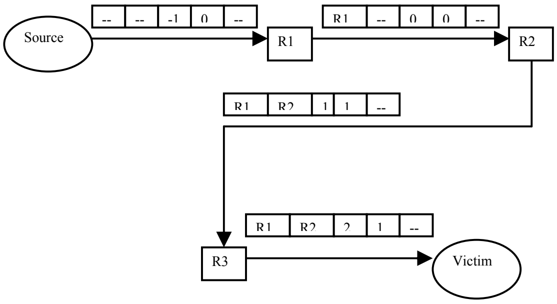


Figure 9. Packet received from source to victim with the efficient encoding

This modified algorithm is named as Efficient probabilistic packet marking algorithm and it is shown in figure 10.

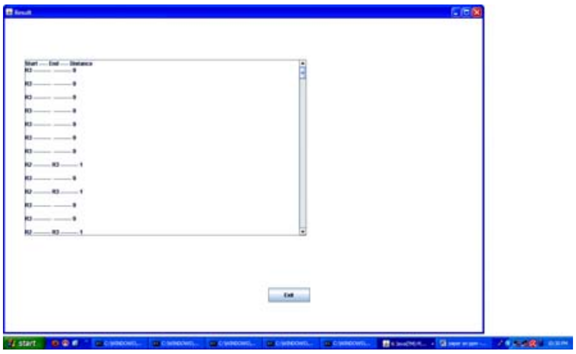
Marking procedure at router R

```
for (each packet w received by the router)
{
    generate a random number x between [0..1);
    if ($x < p_m$ 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 */
    }
}
```

Figure 10. Efficient packet marking procedure

5. EXPERIMENTATION AND RESULT

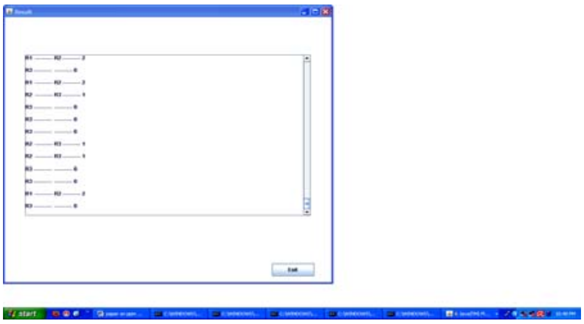
The result we get using the Savage algorithm is as shown in the figure 11. The (R1, R2) edge has been encoded, but R2 can again start encoding if x less than Pm. So in the result we get packets that are encoded with the edges nearer to the victim. In the above we mostly obtain the edge (R2, R3) which is nearer to the victim. If we have more number of routers then the effect of encoding the edges nearer to the victim can easily be observed.



```
Start ---- End ---- Distance
R3 -------- -------- 0
R3 -------- -------- 0
R3 -------- -------- 0
R3 -------- -------- 0
R3 -------- -------- 0
R3 -------- -------- 0
R3 -------- -------- 0
R2 -------- R3 -------- 1
R3 -------- -------- 0
R2 -------- R3 -------- 1
R3 -------- -------- 0
R3 -------- -------- 0
R2 -------- R3 -------- 1
```

Figure 11. Result of the existing Savage packet marking algorithm

Now let us see the result for the effective packet marking algorithm where the edge is encoded only once. From the fig 12 we observed that a part from the edges that are nearer to the victim, there are other edges mainly edges nearer to the source. As an edge once encoded cannot be over written.



```
R1 -------- R2 -------- 2
R3 -------- -------- 0
R1 -------- R2 -------- 2
R2 -------- R3 -------- 1
R3 -------- -------- 0
R3 -------- -------- 0
R3 -------- -------- 0
R2 -------- R3 -------- 1
R2 -------- R3 -------- 1
R3 -------- -------- 0
R3 -------- -------- 0
R1 -------- R2 -------- 2
R3 -------- -------- 0
```

Figure 12. Result of proposed efficient packet marking algorithm**.**

The result after executing the graph reconstruction procedure is as follows:

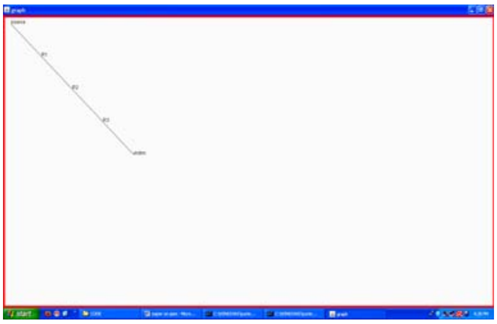


Figure 13. Graph reconstruction.

6. CONCLUSION AND FUTURE WORK

The imminent threats imposed by DoS attacks call for efficient and fast traceback schemes. Some of the desirable features of a good attack traceback scheme are 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.

In this paper we propose a traceback scheme that enjoys the above features. Also, we try to eliminate the major problems of PPM [9]. PPM lacks many of the desirable features mentioned in the beginning. For example, routers that are far away from the victim have very low chance to pass their marking information to the victim because down stream routers overwrite this information, which leads to the loss of valuable marking information written by routers far away from the victim.

Our modified probabilistic algorithm called Efficient Probabilistic Packet Marking algorithm (EPPM) overcome this problem. To conclude, our algorithm (EPPM) is an effective means of improving the reliability of original probabilistic packet marking algorithm.

Our algorithm EPPM is a modified version of PPM algorithm. So EPPM inherits the defects of the PPM algorithm. Further widely distributed attacks and scalability etc will bear future research directions.

REFERENCES

[1] ”CERT Advisory CA-2000-01: Denial-of-Service Developments,”Computer Emergency Response Team, http://www.cert.org/-advisories/-CA-2000-01.html, 2006. International Journal of Network Security & Its Applications (IJNSA), Vol.2, No.3, July 2010 141

[2] J. Ioannidis and S.M. Bellovin, “Implementing Pushback: Router-Based Defense against DDoS Attacks,” Proc. Network and Distributed System Security Symp., pp. 100-108, Feb. 2002.

[3] S. Bellovin, M. Leech, and T. Taylor, ICMP Traceback Messages, Internet Draft -Bellovin- Itrace-04.txt, Feb. 2003.

[4] K. Park and H. Lee, “On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets,” Proc. ACM SIGCOMM ’01, pp. 15-26, 2001.

[5] P. Ferguson and D. Senie, “RFC 2267: Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing,” The Internet Soc., Jan. 1998.

[6] D.K.Y. Yau, J.C.S. Lui, F. Liang, and Y. Yam, “Defending against Distributed Denial-of- Service Attacks with Max-Min Fair Server-Centric Router Throttles,” IEEE/ACM Trans. Networking, no. 1,

[7] C.W. Tan, D.M. Chiu, J.C. Lui, and D.K.Y. Yau, “A DistributedThrottling Approach for Handling High-Bandwidth Aggregates,”IEEE Trans. Parallel and Distributed Systems, July 2007.

[8] D. Dean, M. Franklin, and A. Stubblefield, “An Algebraic Approach to IP Traceback,” ACM Trans. Information and System Security,

[9] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, “Practical Network Support for IP Traceback,” Proc. ACM SIGCOMM ’00,

[10] D.X. Song and A. Perrig, “Advanced and Authenticated Marking Schemes for IP Traceback,” Proc. IEEE INFOCOM ’01, Apr. 2001.

[11] A.C. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio,S.T. Kent, and W.T. Strayer, “Hash-Based IP Traceback,”Proc. ACM SIGCOMM ’01, Aug. 2001.

[12] K. Park and H. Lee, “On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial-of-Service Attacks,” Proc. IEEE INFOCOM ’01, 2001.

[13] M. Adler, “Trade-Offs in Probabilistic Packet Marking for IP Traceback,” J. ACM, Mar. 2005.

[14] V. Paxson, “End-to-End Routing Behavior in the Internet,” IEEE/ACM Trans. Networking Oct. 1997.

[15] M. T. Goodrich, “Efficient Packet Marking for Large-Scale IP Traceback,” in *Proc. Of ACM CCS 2002*, Nov. 2002.

[16] H. Burch and B. Cheswick, “Tracing anonymous packets to their approximate source,” in *Usenix LISA*, 2000.

[17] M. Adler, J.-Y. Cai, J. Shapiro, and D. Towsley, “Estimation of congestion price using probabilistic packet marking,” in *IEEE INFOCOM*, 2003.

[18] R. Stone. Centertrack: An ip overlay network for tracking dos floods. In *Proceedings of 9th USENIX Security Symposium*,August 2000.

[19] S. Lee and C. Shields, “Tracing the source of network attack: A technical, legal and societal problem,” in *IEEE Workshop on Information Assurance and Security*, 2001. tT._y_Ì v kuoµ k/lQ?S_yeT $M_U _k/lqQ8Q8yem7NPOiWTVN Plm7_?¶.M_-aNPM_. ¦ k_ku¶{¨: .ı[.LÊ¦ı.ËT sz _..x.. _ .L._.LÌ

[20] J. Li, M. Sung, J. J. Xu, and L. E. Li, “Large-scale ip traceback in high-speed internet: Practical techniques and theoretical foundation,” in *IEEE SSP*, 2004.

[21] Tao peng, Christopher Leckie and Kotagiri Ramamohanarao. Adjusted probabilistic packet marking for IP traceback. In Proceedings of Networking 2002 Pisa, Italy, May 2002.