IP Traceback With Deterministic Packet Marking

Andrey Belenky and Nirwan Ansari, *Senior Member, IEEE*

***Abstract*—We propose a new approach for IP traceback which is scalable and simple to implement, and introduces no bandwidth and practically no processing overhead. It is backward compatible with equipment which does not implement it. The approach is capable of tracing back attacks, which are composed of just a few packets. In addition, a service provider can implement this scheme without revealing its internal network topology.**

***Index Terms*—IP traceback, security.**

I. Introduction

A GREAT amount of effort in recent years has been directed to the network security issues. In this paper, we address the problem of identifying the source of the attack. We define the source of the attack to be a device from which the flow of packets, constituting the attack, was initiated. This device can be a zombie, reflector, or a final link in a stepping stone chain. While identifying the device, from which the attack was *initiated*, as well as the person(s), behind the attack is an ultimate challenge, we limit the problem of identifying the source of the offending packets, whose addresses can be spoofed. This problem is called the *IP traceback* problem.

Several solutions to this problem have been proposed. They can be divided in two groups. One group of the solutions relies on the routers in the network to send their identities to the destinations of certain packets, either encoding this information directly in rarely used bits of the IP header, or by generating a new packet to the same destination. The biggest limitation of this type of solutions is that they are focused only on flood-based (Distributed) Denial of Service [(D)DoS] attacks, and cannot handle attacks comprised of a small number of packets. The second type of solutions involves centralized management, and logging of packet information on the network. Solutions of this type introduce a large overhead, and are complex and not scalable.

II. Assumptions

The assumptions in this section were largely borrowed from [1]. Some of them were, however, modified to reflect the fact that the scheme is not designed merely for traceback of (D)DoS attacks:

- an attacker may generate any packet;
- attackers may be aware they are being traced;
- packets may be lost or reordered;
- an attack may consist of just a few packets;
- packets of an attack may take different routes;
- routers are both CPU and memory limited;
- routers are not compromised.

III. Deterministic Packet Marking (DPM)

Our proposed algorithm is essentially a packet marking algorithm. We first observe the drawbacks of Probabilistic Packet Marking (PPM), and then try to address them in our proposal.

A. Observations of PPM [1]

In PPM, routers are treated as atomic units of traceback. We propose to treat interfaces as atomic units of traceback. In fact, the IP address of a router means the IP address of one of its interfaces. Making interfaces the units of traceback enables separation of incoming and outgoing packets with respect to a given interface. This will enable packets travelling in one direction to be treated differently from the packets traveling in another direction.

Security issues of PPM schemes arise from the fact that an attacker can inject a packet, which is marked with erroneous information. Such behavior is called *mark spoofing*. Prevention of such behavior is accomplished by special coding techniques, and is not 100% proof. If every packet, which arrives to the victim is ensured to be correctly marked, then the need in those complex and processor intensive encoding techniques will be unnecessary. We propose to ensure that **all** the packets which travel through the network are marked by the routers on the network. In this case, even if an attacker will try to spoof the mark, his spoofed mark will be overwritten with a correct mark.

Finally, we make the following observation about all full-path traceback schemes: in a datagram packet network, the full-path traceback is as good as the address of an ingress point in terms of identifying the attacker. By definition, each packet in a datagram network is individually routed. Since every packet may take a different path from the source to the destination, only the ingress interface on the router closest to the source must be the same. Packets may take different routes even if their source and destination are identical. This may happen for two reasons: due to the unwanted isolation of the network routing, or due to the desired bandwidth management such as load balancing. While it is true that currently, for the most part, the routing on the Internet is stable, it may not be the case in the future.

ISPs may only use public addresses for interfaces to customers and other networks, and use private addressing plans within their own networks. In this case, the usefulness of the full-path traceback becomes very low since information produced for the most part cannot tell the victim much other than

Manuscript received October 28, 2002. The associate editor coordinating the review of this letter and approving it for publication was Prof. D. Petr. This work was supported in part by the New Jersey Commission on Higher Education via the NJI-TOWER project, and the New Jersey Commission on Science and Technology via the NJ Wireless Telecommunications Center.

The authors are with the Advanced Networking Laboratory, Electrical and Computer Engineering Department, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: nirwan.ansari@njit.edu).

Digital Object Identifier 10.1109/LCOMM.2003.811200

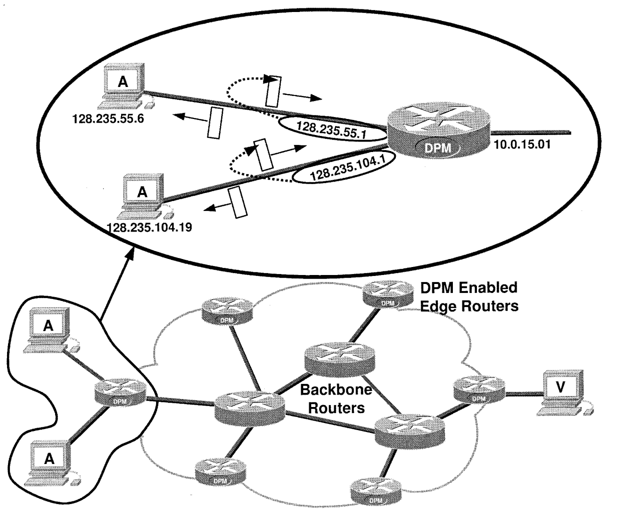

Fig. 1. Deterministic packet marking (DPM).

few IP addresses on the borders between ISPs. Even if this is not the case and public addressing is used within ISPs' networks, ISPs generally feel reluctant to disclose their topologies. Full path traceback schemes reveal topology of all the networks by design. To limit this undesirable behavior, only routers, whose addresses are already known, should implement such schemes.

B. Introduction to DPM

As mentioned above, our algorithm is a packet marking algorithm. The 16-bit Packet ID field and the reserved 1-bit Flag in the IP header will be used to mark packets. *Each* packet is marked when it enters the network. This mark remains unchanged for as long as the packet traverses the network. The packet is marked by the interface closest to the source of the packet on the edge ingress router, as shown in Fig. 1. The mark is a partial address information of this interface, and will be addressed later in Section III-C. The interface makes a distinction between incoming and outgoing packets. Incoming packets are marked; outgoing packets are not marked. This ensures that egress router will not overwrite the mark in a packet placed by an ingress router.

For illustrative purposes, assume that the Internet is a network with a single administration. (The issues of real ISP relationships will be addressed in Section IV-B.) In this case, only interfaces closest to the customers on the edge routers will participate in packet marking. The marking will be done deterministically. Every incoming packet will be marked. Should an attacker attempt to spoof the mark, in order to deceive the victim, this spoofed mark will be overwritten with a correct mark by the very first router the packet traverses.

C. Coding of a Mark

Coding of the mark is one of the ways for PPM schemes to ensure that the mark interpreted by the victim is in fact a valid mark. Since this requirement can now be relaxed we propose here a very simple marking technique.

```
Marking procedure at router R, edge interface I:
  for each incoming packet $w$
    let $x$ be a random number from [0, 1)
    if $x < 0.5$ then
      write $I_{0-15}$ into $w$.ID_field
      write '0' into $w$.flags[0]
    else
      write $I_{16-31}$ into $w$.ID_field
      write '1' into $w$.flags[0]

Ingress address reconstruction procedure at victim V:
  for each packet $w$ from source $S_x$
    if $IngressTbl[S_x] == NIL$ then
      create $IngressTbl[S_x]$
    if $w$.flags[0] == '0' then
      $IngressTbl[S_x]_{0-15} := w$.ID_field
    else
      $IngressTbl[S_x]_{16-31} := w$.ID_field
```

Fig. 2. Pseudo code for the DPM algorithm.

A 32-bit IP address needs to be passed to the victim. A total of 17 bits are available to pass this information: 16-bit ID field and 1-bit reserved Flag. Clearly, a single packet would not be enough to carry the whole IP address in the available 17 bits. Therefore, it will take at least two packets to transport the whole IP address. An IP address will be split into two parts, 16 bits each: the first part—bits 0 through 15, and the second part—bits 16 through 31. With probability of 0.5, the ID field of each incoming packet will be populated with either of those two parts, and then the reserved flag will be set to "0" if it is the first part, and to "1" if it is the second part. It is necessary to introduce this randomness into the scheme so that sophisticated attackers would not send exactly every other packet to the victim, and by doing that creating a situation when only one part of the address is available to the victim. The scheme can be potentially improved by using a nonuniform probability distribution for setting the flag bit so that the probability of having the flag bit of two consecutive datagrams taking different values is maximized.

The coding in the ID Field assumes that there are almost no IP fragments in the Internet. This assumption was made in [1] and is supported by empirical traffic analysis in [2]. According to [2], less than 0.5% of all packets in the Internet are fragmented. This portion of traffic is negligible, but does exist and is a task to be investigated in the future.

D. Formal DPM Description

In this section, we introduce the formal pseudocode for DPM. As seen from Fig. 2, all edge interfaces on all edge routers will place either the first or the last 16 bits in every incoming packet in the ID field, and set the reserved flag to the appropriate value. At the victim, we suggest that the table matching the source addresses to the ingress addresses is maintained. The victim would check to see if the table entry for a given source already exists, and create it if it did not. Then, it would write appropriate bits, depending on the value of the reserved flag, into the ingress IP address value.

The procedures in Fig. 2 are simple and presented here for illustrative purposes. The coding of DPM marks as well as more effective utilization of this information by the victim are open issues, and are among tasks of our future endeavors.

IV. Analysis

In this section, we analyze the performance, topological issues, and benefits of the DPM scheme.

A. Performance Analysis

Deterministic nature of the algorithm ensures that once the ingress point has been identified for a particular source address, it will be correct 100% of the time. By design, DPM prevents mark spoofing.

To ensure successful ingress identification, the victim has to receive two pieces of information: the first 16 bits and the last 16 bits of the ingress interface IP address. Given that packets will be marked with these two packets probabilistically, we are interested in determining how many packets it will take for the victim to gather the complete IP address. It turns out that seven packets are enough on average to be able to generate the ingress IP address with probability of greater than 99%. $(P = 1 - 0.5^7 \simeq 0.9922)$. Similarly, it can be shown that it would take only ten spackets to obtain the ingress interface IP address with probability of more than 99.9%.

B. Topology-Related Analysis

It is unrealistic to assume, of course, that all of the ISPs in the world will engage in DPM. However, it is prudent to assume that even though a given ISP does not participate in DPM, it will honestly inform other ISPs of this fact. It is, therefore, assumed that an upstream ISP knows whether its client ISP implements DPM. If all of the clients in fact do implement DPM, then no action is necessary on behalf of the upstream ISP other than to implement DPM on the interfaces facing its own customers if there are any. If, on the other hand, a client ISP does not implement DPM, it should be treated as a potential attacker by an upstream ISP, and DPM should be implemented on the interface(s) connecting to that client ISP. The range of DPM in this case becomes only as good as a DPM enabled interface on the upstream ISP. However, it should be noted that in most other traceback schemes, if a certain ISP does not wish to participate, traceback through its network will be impossible. More detailed description of the issues of real ISP interactions will be addressed in the future work.

C. Benefits of DPM

The DPM scheme posseses the following merits:

- is easy to implement;
- has low processing and no bandwidth overhead;
- is suitable for a variety of attacks [not just (D)DoS];
- does not have inherent security flaws;
- does not reveal internal topologies of the ISPs;
- is scalable.

V. Conclusions and Future Work

In this letter, we have introduced a new approach to IP traceback called DPM. The approach effectively addresses shortcomings of existing techniques. DPM is light, secure, scalable, and suitable for many types of attacks. In addition, it does not reveal the topologies of ISPs, which implement DPM—this is desirable.

Several issues of DPM were not discussed in this letter. They will be investigated and reported in the near future. For example, to address the fragmentation/reassembly problem, the DPM-enabled interface can suspend the random behavior in assigning the bits to the ID field. The ID field for all fragments of a given series has to be assigned the same address bits. By doing so, the destination would be able to successfully reassemble the original fragmented datagram.

Another modification to the basic approach will be aimed to address the fact that an IP source address can be changed by the attacker during the attack. Though the marks in DPM cannot be spoofed, frequent spoofing/changes of the source address with a different value by an attacker may void the DPM's effectiveness. This problem can be solved by making the destination rely only on the marks, which cannot be spoofed. By using a globally known hash function, the destination can verify that the two halves of the ingress address, received in the marks, do indeed belong to the same ingress address without relying on the source address of the packet. This solution will require sending additional marks with hash values, and will somewhat raise the expected number of packets needed for reconstruction of the ingress address.

Furthermore, we also plan to analyze coding techniques, various probability distributions for assigning the ID field and the reserved Flag, topological issues, deployment issues, and the IPv6 implementation.

References

[1] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, “Network support for IP traceback,” *IEEE/ACM Trans. Networking*, vol. 9, pp. 226–237, June 2001.

[2] C. Shannon, D. Moore, and K. Claffy, “Characteristics of fragmented IP traffic on internet links,” in *Proc. SIGCOMM*, 2001, pp. 83–97.