A Lightweight IP Traceback Mechanism on IPv6

Syed Obaid Amin, Myung Soo Kang, and Choong Seon Hong

School of Electronics and Information, Kyung Hee University,
1 Seocheon, Giheung, Yongin, Gyeonggi, 449-701 Korea
{obaid, mskang}@networking.khu.ac.kr, cshong@khu.ac.kr

Abstract. Due to the stateless nature of the internet, it is an intricate problem to determine the source of spoofed attacking packets. To handle this problem we utilize the IP traceback mechanism to know the actual source of an IP datagram. While many IP traceback techniques have been proposed but most of the previous studies focus and offer solutions for DDoS attacks done on IPv4 environment. Significant difference exists between IPv4 and IPv6 Networks. Therefore, the mechanisms for tracing the origin of attacks in IPv6 networks have abundant differences from those of IPv4 networks. In this paper, we proposed an efficient Lightweight IPv6 traceback algorithm. We also discussed the problems related to previously proposed IPv4 traceback schemes and presented the Error Correction and Modified Path MTU calculation algorithm to make our scheme more practical.[1]

1 Introduction

Every coming day comes with a threatening element for the cyber world. One of them is 'Bot' (also called Botnet) that get into the spotlight recently [12]. Either 'Bot' can be a personal information leakage spyware or DDoS attack agent. In addition, many kinds of DDoS attacks like Smurf [13], UDP flooding [14] and TCP SYN [15] flooding are on the top of cyber attacks' list published by KrCert [11].

To deter these current threats, technologies like Intrusion Detection System (IDS), Intrusion Prevention System (IPS) and the Firewalls are good solutions. But in reality, prevention of all attacks on the internet is nearly impossible and the situation gets more worse due to anonymous nature of IP protocol i.e. an attacker may hide its identity if he wants to. Moreover, the routing decisions are taken on destination addresses and none of the network unit makes sure the legitimacy of source address. Therefore, when prevention fails, a mechanism to identify the source(s) of the attack is needed to at least ensure accountability for these attacks and here we need the traceback techniques.

The elements that were threatening for IPv4 networks can also be intimidating for the future IPv6 network. To cope with IPv6 networks, we need to modify IPv4's traceback technologies to be suited to IPv6 network. The reasons behind this

[1] This work was supported by MIC and ITRC Project. Dr. CS Hong is the corresponding author.

X. Zhou et al. (Eds.): EUC Workshops 2006, LNCS 4097, pp. 671 – 680, 2006.

amendment are the technological differences between these two network-layer protocols for instance, change in header size or fragmentation mechanism.

As mentioned before, the goal of traceback scheme is to identify the true source of a datagram. To achieve this task we try to pass the info of a packet or a path taken by a packet to the victim. So far, the path information is sent to the victim by following methods.

- **Packet Marking:** Routers probabilistically or deterministically mark path information in packets as they travel through the Internet. Victims reconstruct attack paths from path information embedded in received packets. Packet marking techniques can be subdivided in Deterministic Packet Marking (DPM) [10] and Probabilistic Packet Marking (PPM) [2, 3, 4, 9]. Our algorithm lies under the PPM category.
- **Messaging:** Routers probabilistically send ICMP messages, which contain the information of forwarding nodes the packet travels through, to the destination node. Victims reconstruct attack paths from received ICMP messages [1].
- **Packet Digesting:** Routers probabilistically or deterministically store audit logs of forwarded packets to support tracing attack flows. Victims consult upstream routers to reconstruct attack paths [5, 8].

In this paper, we provide some groundbreaking work on PPM Traceback for IPv6 network. The rest of this paper is articulated as follows: Section 2 outlines our proposed technique along with error correction and modified Path MTU algorithm. Section 3 provides the simulation results. In section 4, we describe related work and finally, we summarize our findings in Section 5.

2 Proposal

2.1 Proposed Lightweight IP Traceback Algorithm

This section will discuss the proposed marking algorithm. Marking a packet in IPv6 networks requires a brief analysis of IPv6 header so we can efficiently pass the information to the victim. Along with the compulsory basic header in an IPv6 datagram, one or more extension headers may appear before the encapsulated payload. The complete discussion of these headers is out of scope of this paper and discussed in detail in [16].

Our approach comes under the PPM category and lightweight in a sense that we share the load of packet marking among multiple routers. The router en route probabilistically marks the incoming packets with the Global unicast IPv6 address of that router. The marking algorithm is given below

```
Let q is the marking probability of Router R:
Marking procedure at router R:
    for every packet p
          let x be a random number from [0..1)
          if x < q,
                write R.address into p.node
```

We use Hop-by-Hop Header to store a mark the reasons are two fold; first, the Hop-by-Hop option is processed by every router en-route. Second, it provides the larger space to store a mark as shown in Figure 1. Proposed option in Hop by hop option header is shown in Figure 2.

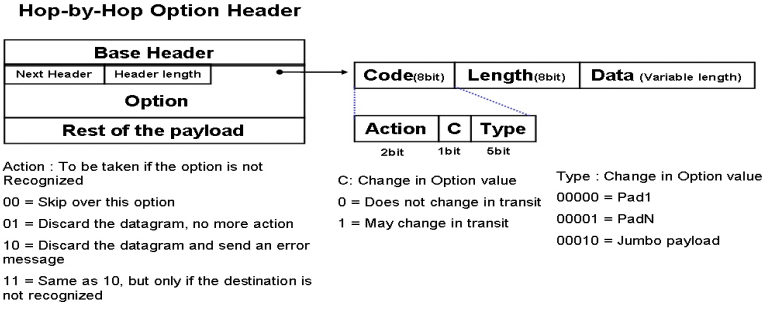

Fig. 1. Structure of Hop-by-Hop Option Field

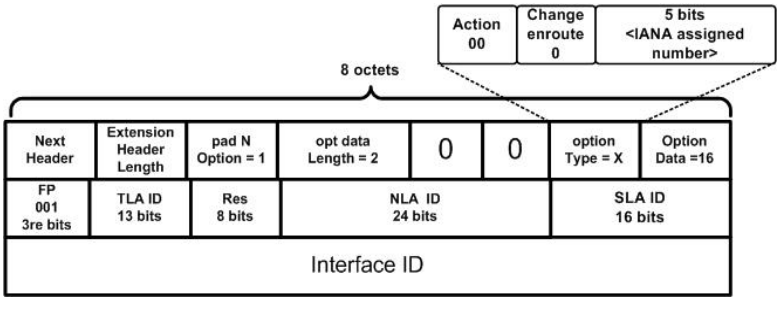

Fig. 2. Proposed Marking Field

Use of extension headers gave us great flexibility to pass the information to the victim. As we mark the packet with complete address, our scheme is not vulnerable to state explosion problem [7], which makes the PPM traceback mechanism proposed in [2] an another DoS type attack. We use these mark packets to construct the reverse routing table from victim to attackers. For this purpose, on victim side, we proposed a data structure called Reverse Lookup Table (RLT) comprises the following fields.

- **Address of Marking Point (AOMP):** Global unicast address of the interface that marked the packet.
- **Distance:** It's a distance of marking router from victim. This can be easily calculated by giving AOMP to any Trace route command from victim side.
- **Last Hop:** It is the address of the last hop from the victim to the attacker. It is used to build the reverse routing table from the victim to the attacker. The method to populate this field is discussed below in detail.

2.2 Traceback

Following steps will be taken to complete the traceback.

1. The victim will sort the RLT by distance field; as shown in figure 3.
2. Observe the discontinuity in distance field and apply the error correction algorithm (discussed later) to find the missing nodes.
3. Finally, victim will resolve the last hop field to complete the RLT.

The resultant sorted tuples of routers can provide a complete path from Victim to attacker.

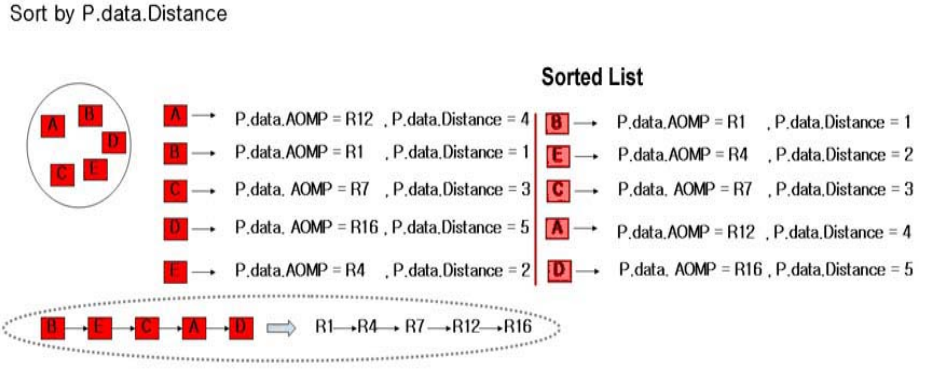

Fig. 3. Reconstructed path using AOMP value and Distance value

Our algorithm works under the assumption that victim is in DDoS attack so the number of evading packets would be sufficient to provide the information of all routes. However, it is quite practical the victim does not have complete route information of the attacker. For this purpose, we introduce the Error Correction Algorithm.

2.3 Error Correction Algorithm

Error correction algorithm is used when the victim is unable to construct a path due to probabilistic nature of marking. In today's internet, when any node sends a datagram to a node on an outlying network, it is not aware of the exact path that this datagram will take; it only has information of a router it is connected to. That router, in turn, looks at the IP address of the destination and decides where the datagram should next "hop" to. This process continues until the datagram reaches the destination host's network. This process is called *Next Hop Routing*. In general, the routing tables have tuples of *Destination address, Prefix length, Next hop* and *administrative weight* there could be other fields but all of these fields are

implementation specific. We utilized the next hop field of routing table to get the missing node as shown in Figure 4.

In this example, after sorting the AOMP by distance field the traceback agent on victim side will observe the discontinuity after the field where distance is 2 and sends a *"routing table request"* message to the router next to the missing node with a destination address of victim, in our case this is R2. R2, in turn, will check it routing table and send the next hop field to the victim's agent. In case of multiple paths from R2 to R3, these records could be multiple.

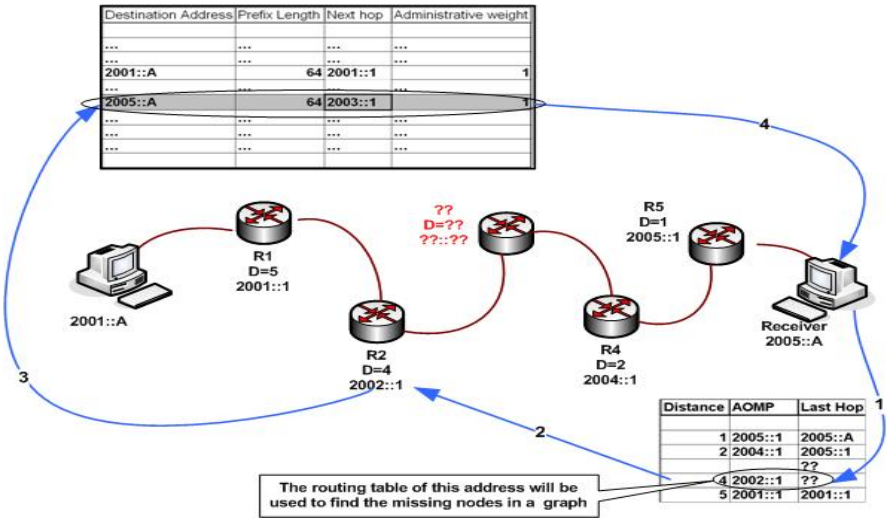

Fig. 4. Error Correction algorithm steps

Error correction algorithm can also be used to fill the last hop field during traceback. This problem is illustrated in following Figure 5. In this figure, R6 is 4 hops away from victim and as there are two entries for D=3 in RLT, victim cannot identify that whether R6 is connected to R4 or R5. In this case, the victim send routing table request message to the router R6 that is next to the routers R4 and R5 to check which of the nodes in R4 and R5 is its next hop, provided victim as a destination. After looking up its Routing table, R6 will inform the Victim that it is connected to R4 instead of R5. The victim then stores the R4 address in "Last hop" field of RLT and continues the process repeatedly for R4 and R5 until he gets absolute path for every D.

Although, it seems that error correction algorithm only works with static routing. It is true up to some extent, but not always. Because, even in dynamic routing routes are not changed that much frequently and may maintain the same state for a long period. Even then, we agree that this is a drawback of error correction algorithm and it is a task to be done and left for our future work.

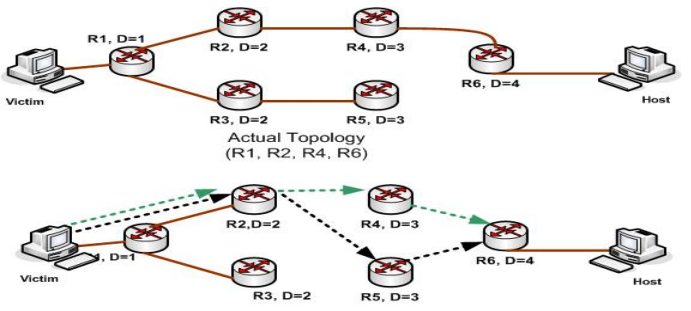

Fig. 5. The victim agent is unable to recognize the that whether R2 is connected to R4 or R5

2.4 Fragmentation Problem

In IPv6 Network, intermediate routers are not able to do fragmentation. Therefore, IPv6 hosts uses Path MTU (PMTU) algorithm before sending a packet to avoid the fragmentation problem. The problem will occur when the sender sends the packet equal to the size of Path MTU then we cannot not add hop-by-hop option header. Because marking the packet with extra 20 bytes will increase the size of packet than PMTU, and since intermediate routers cannot do fragmentation, the packets will be dropped. To eliminate this problem we proposed the modified PMTU algorithm.

The problem discussed above is carried off by subtracting 26 bytes (20 for mark information and 6 for padding) from the discovered PMTU as shown in Fig. 6; so marking of a packet will not increase the size than PMTU and the packets would not be dropped.

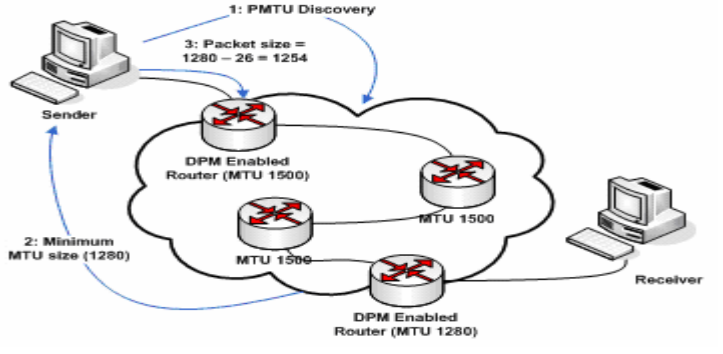

Fig. 6. Modified Path MTU Discovery algorithm

3 Simulation Results

For evaluation, we are interested in the number of packets required to get the full path to the victim. We implemented IP traceback simulator in java for performance

Table 1. Environment parameter

Environment parameter	Value
The number of hop in the path	15
Type of packets that pass by each router and carry different source addresses.	250
The uniform marking probability	1/2000

evaluation, as it is hard to implement the traceback scheme on real networks and not a good support for IPv6 networks in current network simulators. We evaluate the performance by using following environmental parameters given in Table 1.

3.2 Result and Analysis

Fig. 7 shows the expected number of packets required to construct a complete path, when the number of attackers is 50 and 100 calculated by following equation (1) taken from [2]. Here *k* is number of attackers, d is total number of hops and p is the marking probability.

$$E(N) = \frac{k * \ln(k * d)}{p(1-p)^{d-1}} \tag{1}$$

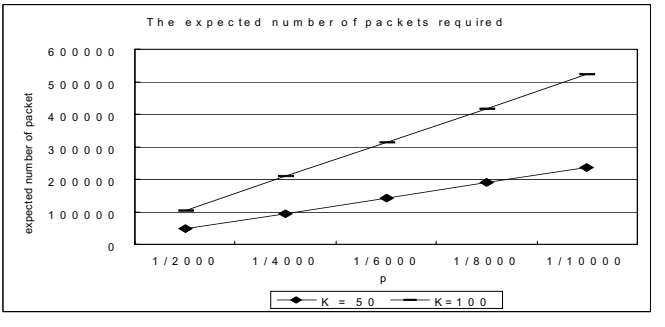

Fig. 7. The expected number of packets required in path reconstruction

In graph shown in Figure 8, the marking probability is 1/2000, the total hop count is 15, and number of packets with different source address passing through each router is 250. The graph represents the average number of routers required to complete the traceback.

In graph shown in Figure 9, the marking probability is 1/2000, the total hop count is 15, and number of packets with different source address passing through each router is 250. The graph represents the average number of routers required to complete the traceback.

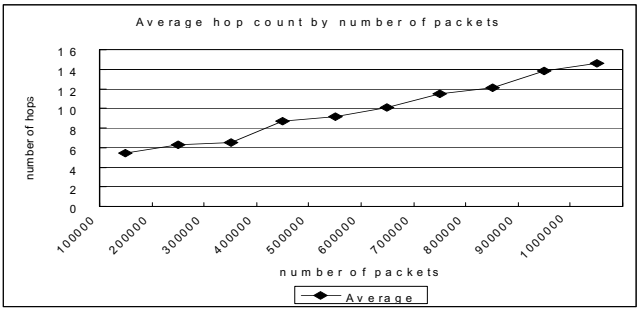

Fig. 8. Reconstructed average hop counts by the number of acquired packets

4 Relevant Work

4.1 Probabilistic Packet Marking

So far, all of the Probabilistic Packet Marking (PPM) algorithms are proposed for IPv4 networks. In Probabilistic Packet Marking [2][3][4] intermediate routers mark packets randomly with any given probability that pass through them with their addresses or a part of their addresses. The victim can reconstruct the full path with given marked packets. This scheme was improved in several different ways; some of them introduced improved coding methods and security. All of the IPv4 PPM algorithms suffered by the space limitation of IPv4 header. Therefore they have to utilize encoding or fragmentation of intermediate router's address. The encoding of each and every packet of course degrades the routing performance while fragmentation of address in small chunks may lead to state explosion problem that is discussed in [7]. As a result, none of the PPM traceback technique has been adapted for the practical work or implementation so far.

4.2 ICMP Traceback

ICMP or iTrace traceback [1] scheme lies under the messaging category. Every router on the network is configured to pick a packet (1 in every 20,000 packets recommended) and generate an ICMP traceback message heading for to the same destination as the selected packet. The ICMP message contains the next and previous hop information, timestamp and as many bytes of the traced packet as possible. The time to live (TTL) field is set to 255, and is then used to identify the actual path of the attack.

This scheme *can* also be deployed on IPv6 networks and presents a very expandable technology if implemented with encryption and key distribution schemes. However, the additional traffic generated consumes a lot of bandwidth even with very low frequency (1/20,000). Without encryption, an attacker can inject false ICMP traceback messages. In addition, ICMP traffic is filtered in many organization to avoid several attack scenarios which make iTrace not that much useful.

4.3 Hash Based IP Traceback

Hash-based traceback [5][6] is officially called Source Path Isolation Engine(SPIE). In hash-based traceback, specialized router (known as data generation agents (DGAs)) confines partial information of every packet that passes through them in form of hash. The network is partitioned logically into regions and each region is controlled by SPIE Collection and Reduction Agents (SCARs) that is connected to all DGAs. The SPIE Traceback Manager (STM) is a centralized unit that communicates to IDSs of the victims and SCARs. In other words, victim starts the traceback through STM.

This technique is very effective and capable to identify a single packet source as well as, according to best of our knowledge, the only scheme which also has solution for IPv6 networks [8]. This scheme, on the other hand, is very computational and resource intensive because tens of thousands of packets can traverse a router every second, the digested data can grow quickly to an enormous size, which is especially problematic for high-speed links.

5 Conclusion

In real world, prevention of all attacks on the Internet is nearly impossible. When prevention fails, a mechanism to identify the source(s) of the attack is needed to at least ensure *accountability* for these attacks and this is the motivation for designing IP traceback schemes. These schemes were not adapted widely for IPv4 networks. One of the main reasons was degradation in routing performance, as encoding should be applied to pass the path information through a limited space IPv4 header.

This paper, contribute in a way that it is the first PPM algorithm for IPv6 network. The extension headers gave us great flexibility to pass the path information to the victim and since the information of routers are not distributed in different fragments as proposed in [3]; our scheme is not affected by the state explosion problem that is discussed in [7]. In future, we will evaluate the other traceback techniques like DPM for IPv6 networks and compare the results with this given solution. We are also improving Error Correction Algorithm so that it can work with dynamic routing.

References

[1] Belenky, A. and Ansari, N. “On IP Traceback," IEEE Communications Magazine, Volume 41, Issue 7, July 2003

[2] S. Savage et al., "Network Support for IP Traceback," IEEE/ACM Trans. Net., vol. 9, no. 3, June 2001, pp. 226-37.

[3] Dawn X. Song and Adrian Perrig, “Advanced and authenticated marking schemes for IP traceback,” in Proceedings IEEE Infocomm 2001, April 2001

[4] K. Park and H. Lee, "On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack," Tech. Rep. CSD-00-013, Department of Computer Sciences, Purdue University, June 2000.

[5] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, B. Schwartz, S. Kent, and W. Strayer. Single-packet IP traceback. ACM/IEEE Transactions on Networking, Dec.2002.

[6] Aljifri, H. "IP traceback: a new denial-of-service deterrent" Security & Privacy Magazine, IEEE , Volume: 1 , Issue: 3 , May-June 2003 Pages : 24 - 31
[7] Marcel Waldvogel, "GOSSIB vs. IP Traceback Rumors", 18th Annual Computer Security Applications Conference (ACSAC '02).
[8] W. Timothy Strayer, Christine E. Jones, Fabrice Tchakountio, and Regina Rosales Hain, SPIE-IPv6: Single IPv6 Packet Traceback, Local Computer Networks, 2004. 29th Annual IEEE International Conference on 16-18 Nov. 2004 Page(s):118 – 125.
[9] Micah Adler, "Tradeoffs in probabilistic packet marking for IP traceback," in Proceedings of 34th ACM Symposium on Theory of Computing (STOC), 2002.
[10] A. Belenky and N. Ansari, .On IP traceback,. IEEE Communications Magazine, vol. 41, no. 7, July 2003.
[11] http://www.krcert.or.kr/
[12] http://en.wikipedia.org/wiki/Botnet
[13] http://en.wikipedia.org/wiki/Smurf_attack
[14] http://en.wikipedia.org/wiki/UDP_flood_attack
[15] http://en.wikipedia.org/wiki/SYN_flood
[16] S. Deering, R. Hinden, Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, IETF, December 1998.