Experience with SPM in IPv6

Mingjiang Ye, Jianping Wu, and Miao Zhang

Department of Computer Science, Tsinghua University, Beijing, 100084, P.R. China
yemingjiang@csnet1.cs.tsinghua.edu.cn
{zm,jianping}@cernet.edu.cn

Abstract. The lack of source IP address checking makes it easy for the attackers to spoof the source address. Spoofing Prevention Method (SPM), as a prospective candidate to be deployed in the Internet, is a newly proposed scheme to solve this problem. However, there is no work on SPM prototype system. In this paper, we present our experience in achieving a prototype system for SPM. In addition to realizing the basic idea of SPM described in the original paper, we make the following three contributions: First, the detail design is made for the whole SPM system architecture and detail mechanisms. Second, several important issues for SPM system are addressed, e.g., how to carry the key required by SPM, the MTU problem, etc. Third, a prototype system is made and some experiments are done with this prototype system.

Keywords: Source Address Spoofing, Spoofing Prevention, Security.

1 Introduction

The Internet is a decentralized system which basically provides best effort, packet-based data forwarding service. In most cases, the source IP address in the IP packet is not checked in the forwarding process. In the Spoofer Project [1], the authors found that approximately one-quarter of the observed addresses, network address blocks and Autonomous Systems (AS) permit full or partial spoofing. The attacks employing source address spoofing remain a serious concern. Source address spoofing is utilized by some DDOS attacks, such as TCP SYN flooding attacks [2], and smurf attacks. The lack of source address validation provides hackers with anonymity, as it is much harder to trace back the source of the attack facilitated with source address spoofing.

Existing schemes to handle IP source address spoofing include [9]: (1) Tracing back the source of the forged packets with the cooperation of routers [3,4]. (2) Filtering forged packets online [5,6,7,8,9]. (3) Using cryptographic authentication, such as IPSec [10].

Though a lot of methods have been proposed, few of which are really adopted in the Internet. There are two important reasons: one is the lack of incentive to deploy. Most of the proposed methods do not bring direct benefit to the ISPs which deploy them. For example, Ingress filtering can only prevent the hosts in an ISP from sending spoofed traffic to other ISPs, but it cannot prevent receiving spoofed traffic from other ISPs.

SPM [6] is a newly proposed method to defense against the spoofed attacks, and it is good at providing incentive for both deployment and incremental deployment. The users in the networks who deploy SPM have direct relative benefits from it, so network operators have strong incentive to implement it. Beside that, even if deployed by only a fraction of the Internet networks, SPM still has significant benefit. These two properties make SPM an attractive solution to the problem of source address spoofing.

SPM filters spoofed packets by checking a unique temporal key, which is associated with each ordered pair of source destination AS (Autonomous System) networks. Each packet leaving a source network S is tagged with the key $K(S, D)$, where D is the destination network. Upon arrival at the destination network, the key is verified and removed.

In the original paper for SPM, the basic idea of SPM is described, and the efficiency of SPM is analyzed. However there is no detail specification for the detail mechanisms and prototype system for SPM. There are also some important issues that are critical to realize SPM in the real world, which have not been discussed in the original paper.

The contribution of this paper is making investigation on these un-addressed issues in implementing and deploying SPM. Firstly, the whole SPM system architecture and detail mechanisms are presented. The functions of SPM are split into three components: register server, AS control server, and AS edge router. Secondly, several important issues for SPM system are addressed. Some issues are related to carrying the key required by SPM in the IP packets. Some issues are related to deploying SPM in a transit AS. The mechanism for managing the SPM key is also discussed.

We made a prototype system, and have done some experiments with this prototype system. The result demonstrated that SPM is capable of preventing the spoofed traffic and working seamlessly with the existing network mechanisms.

2 System Overview

In the SPM paper, the authors made some discussion on how to construct the SPM system. Here a concrete design of the SPM system is proposed as is shown in figure 1, which consists of three components.

- Register Server (RS). RS acts as a rendezvous of the SPM system.
- AS Control Server (ACS). ACS in one AS negotiates SPM key and exchange the AS-Prefix with ACS in other AS. ACS is also responsible for configuring the filter in the AS edge routers.
- AS Edge Router (AER). AER is in charge of adding AS key to each outgoing packet, and verifying the AS key of the incoming packets.

All ASes deployed SPM constitutes a SPM Alliance for protecting against the packet with spoofed source address. Member ASes in the SPM Alliance build up a trust relationship with each other via RS. For each pair of ASes in the SPM Alliance, the negotiation of the key and the exchange of AS-Prefix information are handled by ACS in each AS.

Fig. 1. System Architecture

3 Issues in Designing SPM System

In designing the prototype system for SPM, we find some issues that have not been fully addressed before. They are critical to realize SPM in the real world. Due to the limitation of space, only several important issues are presented in this section.

3.1 How to Carry the Key

It is required by SPM mechanism that a key should be carried in every IP packet. Here how to carry the key in the packet is considered more concretely.

Since there is little space left in the basic IPv6 header, it is necessary to make use of extension header. There are two ways to carry the key with extension header: designing a new type of extension header, or designing a new option in the hop-by-hop extension header. A new extension header is easier for routers to deal with, but it will not be compatible with the current on-shell IPv6 routers. It is expected that SPM will be deployed in an incremental and graceful way, so the focus is put on the second choice.

Fig. 2. IPv6 Hop-by-Hop Options

A new type of Hop-by-Hop Options header [11] is designed for SPM. The option type of this new option is 00001100 in binary format. The highest-order two bits 00 specify that the routers will skip over this option and continue processing the header if this option can not be recognized. The third-highest-order bit 0 specifies that the Option Data can be changed en-route to the packet's

final destination. In current implementation, the option type number is set to 01100, which is not used by other option headers.

The new option header of SPM is compatible with end to end authentication mechanisms because the edge routers of the receiving AS will remove the option header from the packets.

3.2 The MTU Problem

The MTU problem arises from the insertion of SPM key into the IP packet. With the additional SPM key, the whole packet length may exceed the path MTU.

The scenario of MTU problem is depicted in figure 3. The path MTU of the link between router1 and router2 is 1400 bytes, and the path MTU of the link between router2 and router3 is 1320 bytes. When a host is in SPM, AS sends out a packet with 1320 bytes, the edge router of the AS1 tags a 12 bytes long SPM option header with key to the packet. The length of the packet becomes 1332 bytes. Then the packet arrives at router2. Router 2 finds that the length of the packet exceeds the link MTU, and it sends back an ICMP Packet-Too-Big packet with the MTU 1320 bytes. When the host receives the ICMP packet, it may be confused since it sent a packet not exceeding 1320 bytes.

Fig. 3. MTU Problem

As to this problem, a black hole would be set up. The host keeps sending 1320 bytes long packets and it keeps receiving the Packet-Too-Big notification which indicates that the packets are dropped because it is larger than 1320 bytes.

To solve the problem mentioned above, a mechanism is designed to be installed at the AS edge routers to capture and modify the incoming ICMP Packet-Too-Big packets. All the incoming Packet-Too-Big notification packets destined to the local AS will be processed. If the original packets are sent from the local AS to another AS in the SPM Alliance, the MTU value in Packet-Too-Big packets will be modified to a smaller value to reserve the room for the SPM key. For example, in figure 3, the MTU value in ICMP6 packet is modified from 1320 bytes to 1308 bytes.

3.3 Expression for AS-Prefix Ownership

In the SPM, it is necessary to express the ownership of prefix for each member AS of the SPM Alliance. Some discussion should be made on this topic, since it is not as simple as it seems to be.

The expression is very simple for a stub AS. The AS owns all the prefixes assigned to it. However, it is more complex for a transit AS. A fraction of the assigned address of one transit AS may belong to another AS. For example, in figure 4, one multi-homing stub AS3 may have global AS number. And it is allocate a small fraction of address from the provider AS1.

Fig. 4. Multihoming

So the expression for AS-Prefix ownership should be considered more carefully. Two situations are discussed: (1) AS1 and AS3 are both in SPM alliance, (2) AS1 is in the alliance while AS3 is not.

For AS1, it should be explicitly announced that the address space a.b.c.0/24 is not owned by it. If AS3 does not belong to the alliance, ASes in alliance will only receive the announcement for a.b.c.0/24 from AS1. The address space will be marked as non-protected address. If AS3 belongs to alliance, it will announce the ownership for a.b.c.0/24. ASes in alliance receive both announcements for a.b.c.0/24 from AS1 and AS3, so they can conclude that the address space a.b.c.0/24 is owned by AS3 while other parts of a.b.0.0/16 are owned by AS1.

According to the prefix table of AS2 in situation, only AS1 belongs to SPM alliance is shown in table 1. An entry in table will explicitly mark the a.b.c.0/24 as non-protected space. Also, the source of an entry is recorded for update since AS2 may receive new announcement for a.b.c.0/24 in the future if AS3 joins the SPM alliance.

Table 1. AS-Prefix Table

IP address prefix	AS number	Protected address	Announcement Source
x.y.0.0/16	2	Yes	AS2
a.b.0.0/16	1	Yes	AS1
a.b.c.0/24	N/A	No	AS1
x.y.0.0/16	2	Yes	AS2

The longest prefix matching should be used in searching the AS-Prefix table to distinguish the entries such as a.b.c.0/16 and a.b.c.0/24.

3.4 Key Management

Key management is a very important issue for SPM. Our work focused on two points: key negotiation and key switching.

Key Negotiation. The key negotiation happens between the ACS of a pair of ASes in the SPM Alliance. In the original paper, key negotiation is proposed to be sender-driven. The sender AS initiates the key negotiation and the key is generated at the sender AS. Considering the issue of synchronization and the incentive of enabling SPM, receiver-driven key negotiation is suggested. It gives more decision power to receiver AS instead of sender AS.

With receiver-driven scheme, the receiver AS can decide the policies of key management, including:

- (1) Whether to enable the SPM anti-spoofing function with some AS in SPM alliance or not. It is not necessary to enable SPM anti-spoofing function between all peer ASes in the SPM Alliance. An AS can make decision based on the situation.
- (2) Life cycle of the keys for different ASes. For one AS, it can choose different time interval of changing the key for different peer AS.

Key Switching. Key switching is another important issue. When one AS change the key tagged in the packets to another AS, the packets tagged with old key and the packets tagged with new key may coexist in the network at the same time. To avoid dropping those packets tagged with old key, the AS-In Key table keeps both the old key and new key to check the key of the incoming packets, as is shown in Table 2.

Table 2. AS-in Key Table

AS number	Old key		new key	
	Value	Status	Value	Status
M	FE:12:34:CA:89:76:32:45	Valid	32:54:81:29:FF:00:60:21	Valid
N	89:12:34:89:BC:76:FE:3E	Invalid	22:33:65:78:24:70:AB:C0	Valid

The packets tagged with old key should not be allowed infinitely. After having negotiated the new key, the old key should be set to be invalid after a period of time. The length of this period is a parameter which will control how long the old key will be invalid after the new key has been used. Two minutes is a suggested value since it is enough for the packets with old key disappear in the network.

4 Experiment

SPM prototype, including registration server function, controller function and data plane function (separate device schema), is implemented in Linux 2.6.

The prototype system is tested in an IPv6 environment to valid the design issue illustrated above. The small test environment showed in figure 5 has five ASes. Three shadow ASes are ASes in SPM alliance, while the other two ASes are normal ASes. AS1 is a transit AS, connected with a multi-homing stub AS3. A fraction address of AS1 is allocated to AS3.

Fig. 5. Experiment Enviroment

The issues mentioned in previous section are tested in the experiment. The experiment results are showed in the following sections. Three types of traffic were generated in the experiment.

- The traffic from AS5 to AS2. It used the spoofed source address belonging to AS4.
- The traffic from AS3 to AS2. It used the spoofed source address belonging to AS1.
- The traffic from AS3 to AS2. It used the spoofed source address belonging to AS4.

The first type of the traffic was filtered inside of the AS5 by Ingress filtering. None of this type of traffic could be observed in the out link of the AS5. The second type of the traffic successful arrived at the board routers of the AS2, but was filtered by SPM. By inspecting the log of SPM, it was confirmed that the traffic was filtered since carrying the wrong key. The third type of the traffic successfully arrived at the victim. Since the AS3 and AS4 did not belong to SPM alliance, the AS2 could not distinguish the traffic from them. But the rate limitation or other measurement can be taken to protect the victim while not infecting the traffic from the ASes in SPM alliance.

The experiment tested the basic function of SPM. SPM can efficiently protect the ASes in SPM alliance from spoofing attack. The spoofed traffic using the source address belonged to ASes in SPM alliance could be identified and filtered.

5 Conclusion

In this paper, we have discussed the basic principle, the advantage and disadvantage of SPM. The original paper have proposed the SPM mechanism and analyzed the benefit of it. But there are a lot of un-addressed issues left in implementing and deploying SPM. These issues are discussed and solved in the paper.

To validate the architecture and the important issues discussed in the paper, a prototype system is implemented. To our best knowledge, it is the first implementation of the SPM. The experiment results demonstrated that the prototype system is capable of preventing the spoofed traffic and works seamlessly with the existing network mechanisms.

In future work, we will focus on the performance issues in data plane and the deployment SPM system in the real world.

References

1. Beverly, R., S. Bauer: The spoofer project: inferring the extent of source address filtering on the Internet. Proceedings of USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI 2005), Cambridge, MA (2005) 53–59
2. C. Schuba, I. Krsul, M. Kuhn, E. Spafford, A. Sundaram, D. Zamboni: Analysis of a denial of service attack on TCP. Proceedings of the 1997 IEEE Symposium on Security and Privacy. IEEE Computer Society, Washington, DC, USA (1997) 208–223
3. A. Yaar, A. Perrig, D. Song: Pi: A Path Identification mechanism to defend against DDoS attacks. Proceedings of the 2003 IEEE Symposium on Security and Privacy. IEEE Computer Society, Washington, DC, USA (2003) 93–107
4. J. Li, M. Sung, J. Xu, L. Li: Large-scale IP traceback in high-speed Internet: Practical techniques and theoretical foundation. Proceedings of the 2004 IEEE Symposium on Security and Privacy. IEEE Computer Society, Washington, DC, USA (2004) 115–129
5. P. Ferguson, D. Senie: Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing. RFC 2267 (2000)
6. Bremner-Barr, A., Levy, H: Spoofing Prevention Method. IEEE INFOCOM Vol. 1, (2005) 536–547
7. C. Jin, H. Wang, K. G. Shin: Hop-count filtering: An effective defense against spoofed DDoS traffic. Proceeding of the 10th ACM International Conference on Computer and Communications Security (CCS 03). ACM Press, New York, USA (2003) 30–41
8. T. Peng, C. Leckie, R. Kotagiri.: Protection from distributed denial of service attacks using history-based IP filtering. Proceedings of the IEEE International Conference on Communications Vol. 1. Anchorage, Alaska, USA (2003) 482–486
9. J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang: SAVE: Source Address Validity Enforcement Protocol. IEEE INFOCOM 2002 Vol. 3, (2002) 1557–1566
10. S. Kent, R. Atkinson: Security architecture for the Internet protocol. RFC 2401 (1998)
11. S. Deering, R. Hinden: Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (1998)