Enable a Trustworthy Network by Source Address Spoofing Prevention Routers: A Formal Description

Jun Bi, Jianping Wu, and Miao Zhang

Network Research Center, Tsinghua University
Beijing, P.R. China, 100084
junbi@tsinghua.edu.cn

Abstract. The lack of verifying source address in Internet makes it easy for attackers to spoof the source IP address. One of challenges of Internet has been recognized is building mechanisms in routers to verify the source address. This paper discusses Source Address Spoofing Prevention (SASP) mechanisms, presents a formal description on SASP network and SASP router, proposes a hierarchical SASP architecture, and presents some design principles of SASP mechanisms.

Keywords: Source Address Spoofing, Source Address Validation, Network Security.

1 Introduction

The fundamental principles of today's Internet are best-effort and destination address based packet forwarding. The lack of verifying source address of IP packets being forwarded through a router makes it easy for the attackers to spoof a source IP address other than the accurate address of the attacking host. [1] shows that approximately 24% of the observed net-blocks, corresponding to 25% the observed autonomous systems (AS) from which spoofing is possible. One of challenges of Internet has been recognized is building mechanisms in routers to verify the source address. Enabling the Source Address Spoofing Prevention (SASP) network is not only helpful to network security, but also helpful to network application, network management and accounting.

In recent years, there have been some efforts in the research community to design mechanisms on fighting against source address spoofing, such as Ingress Filtering based method [2], Traceback based method [3], Incoming Table based method [4], etc. However, these mechanisms are not widely deployed in the Interne, mainly due to two reasons: the incentive for ISPs to deploy these mechanisms is relative low, and the incremental deployment is not well supported.

The motivation of this paper is to present formal definitions and analysis for source address spoofing prevention problem. To reduce the complexity, a source address spoofing prevention network needed to be decomposed through a process of abstraction in network topology. Multiple candidate solutions, strict SASP, moderate SASP, and loose SASP, can be designed for each abstract level.

The rest of this paper is organized as follows: in Section 2, related works are introduced and analyzed; in Section 3, we present the formal models and definitions

X. Zhou et al. (Eds.): EUC Workshops 2006, LNCS 4097, pp. 681 – 691, 2006.
© IFIP International Federation for Information Processing 2006

of SASP network; in Section 4, we propose the hierarchical architecture of SASP; Section 5 indicates the design principles for SASP mechanisms; and section 6 summarizes the paper.

2 Related Works

The existing SASP mechanisms can be classified from different aspects.

2.1 Origin vs. Destination

According to the deployment position, the related mechanisms can be classified into four categories:

1. Deployed in the origin network

The mechanism is deployed in the origin network where the packet is generated. Ingress filter based method [2] is a case for such mechanism.

Preventing source address spoofing at the origin is the most effective. However, it provides little incentive for the deployment, because they are not self-defensive. According to the analysis in [5], the damage reduction rate of ingress filtering is K/N, where K denotes the number of domains that implement the defense, and N is the total number of domains.

2. Deployed in the middle of network

The mechanism is deployed in the middle path of packet transmission. It either filters out packets with spoofed source address, e.g., methods presented in [6][7]; or supports tracing back to the origin of the packet, e.g., methods presented in [3][8][9].

Normally such mechanism is hard to implement the incremental deployment. The incentive for deployment is also a problem

3. Deployed in the destination network

The mechanism is deployed in the destination network of packet transmission, e.g., methods resented in [10][11][12].

This mechanism is self-defensive, and provides incentive for the deployment. However, it is hard to design and implement a perfect mechanism without the cooperation from the origin and the middle of network.

4. Deployed in multiple positions

Some mechanisms require to be deployed at more than one position, e.g., methods presented in [4][5][13].

According to the complexity of SASP, in our opinion, the final solution should be a mechanism that relies on participations of different parts of the network.

2.2 Proactive vs. Reactive

The related mechanisms can be classified into two categories, according to the type of reaction:

1. Reactive mechanisms

The idea of reactive method is to take action after some abnormal traffic is detected. Traceback based methods [3][8][9] fall in this category.

2. Proactive mechanisms

The goal of proactive method is to discard the packet with spoofed source address before it can reach the ultimate destination. Methods that apply packet filtering fall in this category.

Considering the quick response required to handle source address spoofing events, we think that proactive mechanism will be the better choice for the final SASP solution.

2.3 Path/Route Based vs. End-to-End Based

To generate a filtering information database in routers, some SASP information should be exchanged among routers. According to the type of information exchanged among routers, the mechanisms can be classified into two categories:

1. Path/Route based mechanisms

The information is derived from the path that the packet transmitting along, or from the routing information base. Most existing mechanisms fall in this category.

The disadvantage of this mechanism is the complexity of a real network makes it hard for routers to generate the path/route based information.

2. End-to-end based mechanisms

Such mechanisms only rely on end-to-end information (e.g., key) to check the authenticity of the source address. SPM [5] is one example of such mechanism.

The disadvantage of this mechanism is the overhead to negotiate the end-to-end key for each peer, if the total number of peers is relative large.

According to the disadvantage discussed above, we think the final solution will mix both path/route and end-to-end information.

3 Source Address Spoofing Prevention Network

3.1 Definition and Benefits of SASP

From our viewpoint, the source address spoofing prevention network can be described as follows.

1. The source address of each packet is globally authorized and unique.

Enabling the SASP network will make the authentication of a higher layer network entity simplified. The identity of a network entity or application can be designed and mapped based upon source address.

2. A packet with spoofed IP source address won't be forwarded by SASP routers.

Enabling the SASP network will make it impossible for a hacker to launch network attacks with spoofed source address.

3. Each packet will be forwarded only from the authorized location, and can be traced back to the origin accurately.

Enabling the SASP network will help network administrators to trace to the source of network attacks. Moreover, the network billing system can easily map each application packet to a specific user, then bill the usage of that application to the owner of that source IP address, just like bill the telephone usage to the owner of a calling number.

3.2 Formal Models of SASP Network

To formally model the source address spoofing prevention network, we present formal definitions of some preliminary concepts.

Definition 1. Atomic Entity
An Atomic entity, denoted as a∈ A, is an entity which could not be subdivided, where,

A = H ∪ R;

H is the set of hosts: h ∈ H;

R is the set of routers, including network address translators (NAT): r ∈ R.

Definition 2. Interaction Point
An interaction point, denoted as i ∈I, is a location where packet could be received or sent.

The interaction point is used to denote a link between a host and a router, or a link between routers.

Definition 3. Packet
A packet, denoted as p∈ P, is a 3-tuple <s, t, m>, where:

s is the origin of the packet, s ∈ A;

t is the destination of the packet, t ∈ A;

m ∈M denotes a possible mark in a packet used by some spoof prevention; for some methods do not mark a packet, m ∈∅ .

Definition 4. SASP Network
A SASP network *sasp* is a 7-tuple <A, I, P, O, Tr, Fo, Fi>, where:

A is a non-empty set of atomic entities, |A| >1;

I is a non-empty set of interaction points, | I | ⩾1;

P is a set of packets that can be observed at interaction points;

O: A→PowerSet (I) is a set of topology functions associating each entity with a set of interaction points. It denotes interfaces of an entity;

Tr: P→ PowerSet (A×I) is a set of trace functions associating a packet with a set of 2-tuple (a processing entity and a interaction point). It denotes the trace of a packet transmitted in the network;

Fo: R × I→ PowerSet (A) is a set of forwarding functions associating a processing router and an output interaction point with a set of destination address. It denotes the forwarding information base in a router;

Fi: R × I → PowerSet (A) is a set of filtering functions associating a processing router and an input interaction point with a set of source address. It denotes the filtering information base in a router. For a router *r* doesn't use filtering mechanism, fi (a, r) = A.

Definition 5. Strict SASP (SSASP) Network
The *ssasp* = $<A, I, P, O, Tr, Fo, Fi>$ is a strict SASP network, if and only if,

$\forall\ p=<s, t>,$
$\forall <r, i> \in tr(p), i \in o(r), s \in A, t \in A, p \in P, r \in R\ (R \subset A), i \in I, tr \in Tr, o \in O,$
$s \in fi(r, i), fi \in Fi$.

This definition of SSASP means that every host and router can be traced back accurately; and it also means private address space is not allowed because every *s* (including every host) is traceable.

Definition 6. Moderate SASP (MSASP) Network
The *msasp* = $<A, I, P, O, Tr, Fo, Fi>$ is a moderate SASP network, if and only if,

$\forall\ p=<s, t>, s \in R\ (R \subset A),$
$\forall <r, i> \in tr(p), i \in o(r), t \in A, p \in P, r \in R\ (R \subset A), i \in I, tr \in Tr, o \in O,$
$s \in fi(r, i), fi \in Fi$.

This definition of MSASP means that only routers or NAT gateways ($s \in R$) can be traced back, it also means the network allows a private address space behind a NAT.

Definition 7. Loose SASP (LSASP) Network
The *lsasp* = $<A, I, P, O, Tr, Fo, Fi>$ is a loose SASP network, if and only if,

$\forall\ r \in R\ (R \subset A),$
$\forall\ p=<s, t>,$
$\exists <r, i> \in tr(p), i \in o(r), s \in A, t \in A, p \in P, r \in R\ (R \subset A), i \in I, tr \in Tr, o \in O,$
$s \in fi(r, i), fi \in Fi$.

This definition of LSASP means that source address spoofing prevention was enabled only in part of packet transmission paths, for example, was enabled at the edge of destination network.

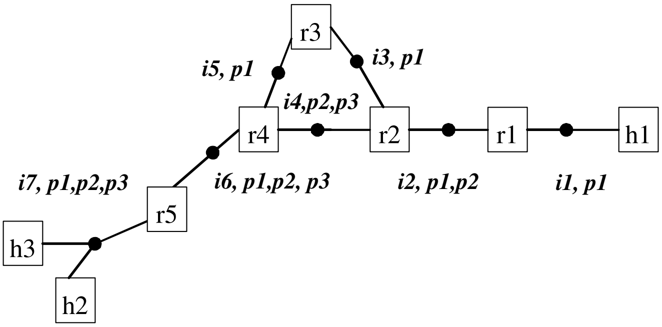

Fig. 1. Example 1 of SASP network

Figure 1 shows an example of the formal model defined above. In this case, the network is defined as <A, I, P, O, Tr, Fo, Fi>, where

H= {h1, h2, h3};

R={r1, r2, r3, r4, r5};

A={ h1, h2, h3, r1, r2, r3, r4, r5};

I = {i1, i2, i3, i4, i5, i6, i7};

P= {p1, p2, p3}, where p1=<h1, h2>, p2=<h3, r1>, p3=<r2, h3>;

O= {o1, o2, o3, o4, o5, o6, o7, o}, where o1(h1)={i1}, o2(r1)={i1, i2}, o3(r2)={i2, i3, i4}, o4(r3)={i3, i5}, o5(r4)={i4, i5, i6}, o6(r5)={i6, i7}, o6(h2)={i7}, o7(h4)={i7};

Tr={tr1, tr2, tr3}, where tr1(p1)={<r1, i1>, <r2, i2>, <r3, i3>, <r4, i5>, <r5, i6>, <h2, i7>}, tr2(p2)={<r5, i7>, <r4, i6>, <r2, i4>, <r1, i2>}, tr3(p3)= {<r4, i4>, <r5, i6>, <h3, i7> };

Fo={fo1, fo2, fo3, fo4, fo5, fo6, fo7, fo8, fo9, fo10, fo11, fo12}, where fo1(r1, i1)={h1}, fo2(r1, i2)={h2, h3, r2, r3, r4, r5}, fo3(r2, i2)={h1, r1}, fo4(r2, i3)={r3}, fo5(r2, i4)={h2, h3, r4, r5}, fo6(r3, i3)={h1, r1, r2}, fo7(r3, i5)={h2, h3, r4, r5}, fo8(r4, i4)={h1, r1, r2}, fo9(r4, i5)={r3}, fo10(r4, i6)={h2, h3, r5}, fo11(r5, i6)={h1, r1, r2, r3, r4}, fo12(r5, i7)={h2, h3};

Fi={fi1, fi2, fi3, fi4, fi5, fi6, fi7, fi8, fi9, fi10, fi11, fi12}, where fi1(r1, i1)={h1}, fi2(r1, i2)={h2, h3, r2, r3, r4, r5}, fi3(r2, i2)={h1, r1}, fi4(r2, i3)={h2, h3, r3, r4, r5}, fi5(r2, i4)= {h2, h3, r3, r4, r5}, fi6(r3, i3)= {h1, h2, h3, r1, r2, r4, r5}, fi7(r3, i5)= {h1, h2, h3, r1, r2, r4, r5}, fi8(r4, i4)={h1, r1, r2, r3}, fi9(r4, i5)= {h1, r1, r2, r3}, fi10(r4, i6)={h2, h3, r5}, fi11(r5, i6)={h1, r1, r2, r3, r4}, fi12(r5, i7)={h2, h3}.

4 Hierarchical Architecture of SASP

4.1 Formal Definitions

From definition 4 and the example, we can figure out that the complete specification of a real SASP network involves a very large amount of information. Attempting to capture information of all aspects in a single description is generally unworkable.

A network can be simplified by establishing a set of models, each aims at capturing one aspect the network, satisfying the requirements that are the concern of some particular group. The *viewpoint* of a network can express concepts and rules relevant to a particular area of concern in terms of which of a network can be described from that viewpoint.

A SASP network can be described as < vpt_1, vpt_2, ... , vpt_i , …vpt_n >, where vpt_i is the i-th level of viewpoint of the network, vpt_1 is called the bottom viewpoint, and vpt_n is called the top viewpoint.

Definition 8. Viewpoint

A viewpoint $vpt_i \in$ VPT is the *i*-th level viewpoint of a SASP network. It is a 8-tuple < E_i , D_i , I_i , P_i , O_i, Tr_i , Fo_i , Fi_i>, where:

E_i is a non-empty set of entities in vpt_i, $| E_i | >1$; at the bottom viewpoint, it is equal to the set of atomic entities;

$D_i = E_{i+1}$ is a non-empty set of domains, $| D_i | \geqslant 1$; The domain of *i*-th level viewpoint is the entity of upper level viewpoint: a domain d_i is a set of entities, $d_i \subset E_i$; and at the top viewpoint, $D_n = \{ E_i \}$;

I_i is a set of interaction points, $| I_i | \geqslant 1$;

P_i is a set of packets that can be observed at interaction points;

O_i: $E_i \rightarrow$PowerSet (I_i) is a set of topology functions associating each entity with a set of interaction points. It denotes interfaces of an entity;

Tr_i: $P_i \rightarrow$ PowerSet ($E_i \times I_i$) is a set of trace functions associating a packet with a set of 2-tuple (a processing entity and a interaction point). It denotes the trace of a packet transmitted in the network;

Fo_i: $E_i \times I_i \rightarrow$ PowerSet (E_i) is a set of forwarding functions associating a processing entity and an output interaction point with a set of destination entity;

Fi_i: $E_i \times I_i \rightarrow$ PowerSet (E_i) is a set of filtering functions associating a processing entity and an input interaction point with a set of source entity. For an entity *e* doesn't participate in SASP operation, fi_i (e, i) = E_i.

In an upper level viewpoint, we only need to consider the source of packets at interaction points between domains. It simplifies the topology, reduces the number of interaction points where packets need to be filtered.

Figure 2 shows an example of a hierarchical SASP network. Most elements of $viewpoint_1$= < E_1, D_1, I_1, P_1, O_1, Tr_1, Fo_1, Fi_1> are analyzed in the last example. There

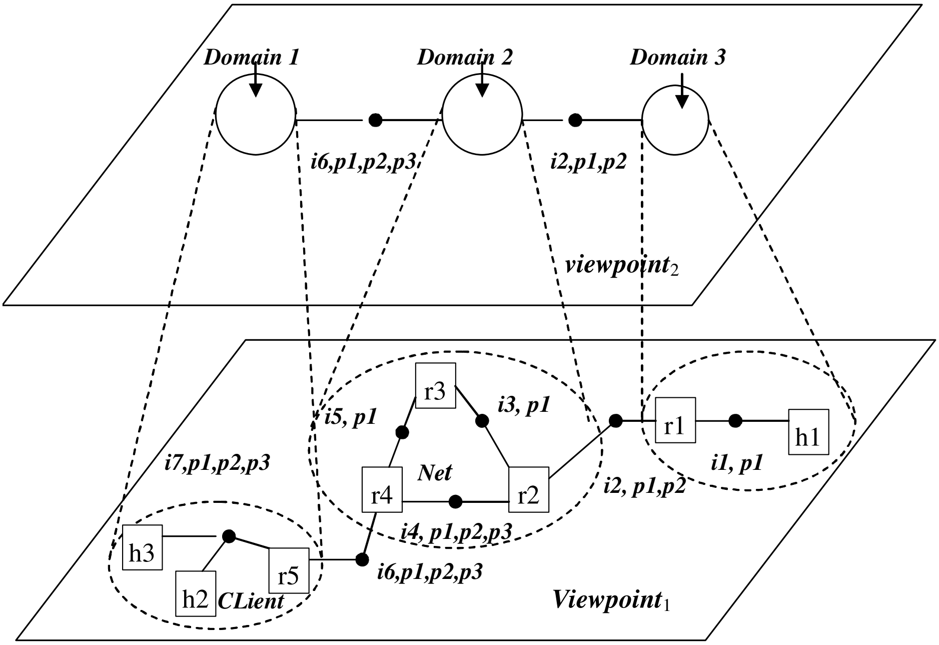

Fig. 2. Example of a hierarchicazl SASP network

are three domains in viewpoint$_1$: D_1 = {d1, d2, d3}, where d1 = {h2, h3, r5}, d2= {r2, r3, r4}, d3= {h1, r1}.

Viewpoint$_2$= < E_2, D_2, I_2, P_2, O_2, Tr_2, Fo_2, Fi_2>, where:

E_2 = D_1 = {d1, d2, d3};

D_2 ={{d1, d2, d3}};

I_2 ={i2, i6};

P_2 ={p1, p2, p3}, where p1=<d3, d1>, p2=<d1, d3>, p3=<d2, d1>;

O_2= {o1, o2, o3 }, where o1(d1)={i6}, o2(d2)={i2, i6}, o3(d3)={i2 };

Tr_2={tr1, tr2, tr3}, where tr1(p1)={<d2, i2>, <d1, i6>}, tr2(p2)={<d2, i6>, <d3, i2>}, tr3(p3)= {<d1, i6> };

Fo_2={fo1, fo2, fo3, fo4 }, where fo1(d1, i6)={d2, d3}, fo2(d2, i2)={d1}, fo3(d2, i6)={d1}, fo4(d3, i2)={d1, d2};

Fi_2={fi1, fi2, fi3, fi4}, where fi1(d1, i6)={d2, d3}, fi2(d2, i2)={d1}, fi3(d2, i6)={d1}, fi4(d3, i2)={d1, d2}.

From example 2, we can see the elements and functions are quit simplified in viewpoint$_2$. Hence we propose the hierarchical SASP architecture to enable SASP in different viewpoints.

Definition 9. Hierarchical SASP (HSASP) Network

The *hssasp* =<vpt$_1$, vpt$_2$, ... ,vpt$_i$, …vpt$_n$ > is a hierarchical SASP network, if and only if,

$\forall$ vpt$_i$= <E_i, D_i, I_i, P_i, O_i, Tr_i, Fo_i, Fi_i>,
$\forall$ $d_{ij} \in D_i$,
$\forall$ p=<s, t >, $s \in d_{ij}$
$\forall$ <r, i> $\in$ tr (p), $i \in o(r)$, $r \in d_{ij}$, $t \in E_i$, $p \in P_i$, $i \in I_i$, $tr \in Tr_i$, $o \in O_i$,
$s \in$ fi (r, i), fi $\in Fi_i$.

The definition of hierarchical SASP means that at each viewpoint, we only enable SASP within a domain and filter packets originated from entities within that domain.

4.2 Real World Considerations

In the real world network, the viewpoint can be considered as 3 levels. At the top level viewpoint$_3$, the entity can be autonomous systems (AS). An AS will decide whether forward or accept a packet at the interaction points between ASes, based on the information if it comes from the correct AS. The prefix address and AS numbers owned by ASes can be used as the $E_{3.}$

At the second level viewpoint$_2$, the entities can be subnetworks including stub-networks, while the domains can be ASes. Some subnetwork, especially the stub-network, usually has the fixed address prefix, which can be used as the E_2. A subnetwork only checks packets originated in the local AS (domain), and the filtering decision is based on the information if the packet comes from the correct subnetwork.

At the bottom level viewpoint$_1$, the entities are hosts, routers, and NATs, while the domains can be subnetworks. A router only checks the packets originated in the local sub-network (domain), the filtering decision is made based on the information if the packet has the correct source address within this sub-network.

The hierarchical SASP used for different scenarios are shown in table 1.

Considering the size of a subnetwork is not large, it will be easier to deploy SASP mechanisms within a subnetwork. Meanwhile, the number of subnetworks in an AS is relative small, thus the SASP deployment among subnetworks within an AS is not complex. The total number of ASes in the Internet is in 10Ks, and it also makes SASP deployment among ASes possible.

Table 1. Hierarchical SASP used for different scenarios

Destination Source	local AS, local sub-network	local AS, other sub-networks	other ASes
local AS, local sub-network	Intra-subnetwork SASP	Intra-subnetwork SASP	Intra-subnetwork SASP
local AS, other sub-networks	Inter-subnetwork SASP	Inter-subnetwork SASP	Inter-subnetwork SASP
other ASes	Inter-AS SASP	Inter-AS SASP	Inter-AS SASP

5 Design Principles

In designing the SASP mechanism, the following design principles should be considered.

1. Incremental deployment

The mechanism should show its benefits even if it is deployed in part of the network. It is impossible to deploy some mechanism all over the Internet in a short time. If there is no benefit from partial deployment, it will be hard to start the deployment.

2. Provide incentive to deploy

The mechanism should have direct benefits to the party who makes investment on the deployment of SASP mechanism. Otherwise there is no enough incentive for the party to invest on the deployment.

3. Consistence of existing protocols and mechanisms

The deployment of new mechanisms should not affect the existing protocols and mechanisms.

In the design of some SASP mechanisms, some data fields (e.g., TTL, IPid, and Flow Label) in the IPv4/IPv6 packet header are utilized to carry some control information. It should not affect the normal protocol operation and should leave space for future protocol design.

4. Manageable

The mechanism should be easy to be managed and configured. The management overhead of new mechanism should be considered.

5. Simple to be implemented

The mechanism should be simple enough to be implemented in routers or hosts.

6. Multi-homing

It should be considered how to support multi-homing.

7. Private address space

If there exists a private address space, then a packet can be only traced back at the source NAT gateway. According to the definition of SASP, the source address of a packet originated in a private address space is not globally unique, it can't be traced back to the origin accurately, and the source address can't be used as the identity of an application. Therefore, seriously consider how to detect and disable NAT is an import topic in SASP study.

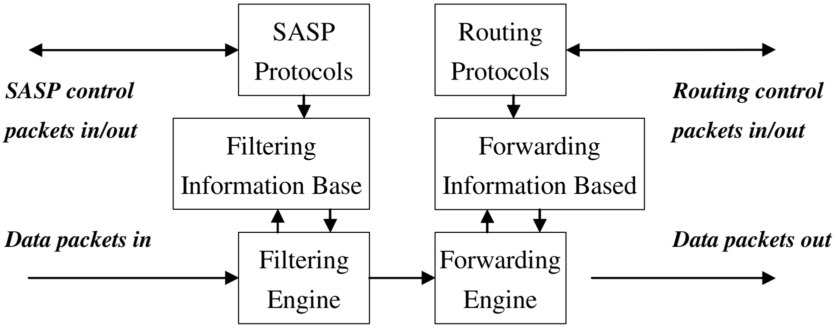
SASP Protocols
Routing Protocols
SASP control packets in/out
Filtering Information Base
Forwarding Information Based
Routing control packets in/out
Data packets in
Filtering Engine
Forwarding Engine
Data packets out

Fig. 3. Architecture of the SASP router

A trustworthy network is implemented by the deployment of SASP router. Figure 3 shows the architecture of SASP router, which contains the following major logical parts:

1. Forwarding information database

This database contains packet forwarding information: Fo: R × I→ PowerSet (A).

2. Filtering information database

This database contains SASP filtering information: Fi: R × I → PowerSet (A).

3. Routing protocols

This module exchanges routing information among routers, then updates forwarding information database.

4. SASP protocols

This module exchanges SASP information among routers, then updates filtering information database.

5. Packet forwarding engine.

This module forwards data packets according to forwarding information base.

6. Packet filtering engine

This module filters incoming data packets according to filtering information base, and pass legal packets to packet forwarding engine.

6 Conclusions

A trustworthy network can be built based upon the deployment of source address spoofing prevention. Recent works proposed different spoofing prevention methods. This paper presents a formal description on SASP network and the formal model will help researchers to clearly state and analyze the problem. This paper also proposes the concepts of strict SASP, moderate SASP, loose SASP and hierarchical SASP, and indicates the design principles of SASP mechanisms. A hierarchical architecture will help to simply the problem, and SASP methods can be deployed within a domain based on the simplified scenario.

Because the current Internet addressing architecture has been formed for a long time, it is quit difficult and not cost-effective to deploy SASP in the current Internet. The IPv6 based next generation Internet might be an opportunity for us to implement a trustworthy network. IPv4/IPv6 transition will also help to build the IPv6 network where end systems are enforced with authorized IPv6 source address, based upon IPv4 infrastructure.

References

1. Beverly, R. and Bauer S.: The Spoofer Project: Inferring the Extent of Source Address Filtering on the Internet, USENIX SRUTI 2005.
2. Ferguson, P. and Senie, D.: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing, RFC 2827, May 2000.
3. Bellovin S.: ICMP Traceback Messages, Internet Draft draft-bellovin-itrace-00.txt, March 2000.
4. Li, J., Mirkovic, J., Wang, M., Reiher, P., and Zhang, L.: SAVE: Source Address Validity Enforcement Protocol, IEEE INFOCOM 2002.
5. Bremler, A. and Levy, H.: Spoofing Prevention Method, IEEE INFOCOM 2005.
6. Park, K. and Lee, H.: On the effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets, ACM SIGCOMM 2001.
7. Park, K. and Lee, H.: On the effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack, IEEE INFOCOM 2001.
8. Savage, S., Wetherall, D., Karlin, A., Anderson, T.: Practical Network Support for IP Traceback, ACM SIGCOMM 2000, 2000.
9. Alex, S., Sanchez, L., Jones, C., Tchakountio, F., Schwartz, B., Kent, S., and Strayer, W.: Single-Packet IP Traceback, ACM SIGCOM 2001, 2001.
10. Jin, C., Wang, H., and Shin, K.: Hop-count Filtering: An Effective Defense Against Spoofed DDoS Traffic, ACM Conference on Computer and Communications Security 2003.
11. Yaar, A., Perrig, A., and Song, D.: Pi: A Path Identification Mechanism to Defend against DDoS Attacks, IEEE Symposium on Security and Privacy 2003.
12. Yaar, A., Perrig, A., and Song, D. : StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense, IEEE Journal on Selected Areas in Communications 2006.
13. Mirkovic, J., Xu, Z., Li, J., Schnader, M., Reiher, P., and Zhang, L.: iSAVE: Incrementally Deployable Source Address Validation, UCLA technical report, 2002.