Internet Engineering Task Force (IETF)
Request for Comments: 7039
Category: Informational
ISSN: 2070-1721

J. Wu
J. Bi
Tsinghua Univ.
M. Bagnulo
UC3M
F. Baker
Cisco
C. Vogt, Ed.
October 2013

Source Address Validation Improvement (SAVI) Framework

Abstract

Source Address Validation Improvement (SAVI) methods were developed to prevent nodes attached to the same IP link from spoofing each other's IP addresses, so as to complement ingress filtering with finer-grained, standardized IP source address validation. This document is a framework document that describes and motivates the design of the SAVI methods. Particular SAVI methods are described in other documents.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7039.

Table of Contents

1. Introduction

Since IP source addresses are used by hosts and network entities to determine the origin of a packet and as a destination for return data, spoofing of IP source addresses can enable impersonation, concealment, and malicious traffic redirection. Unfortunately, the Internet architecture does not prevent IP source address spoofing [RFC6959]. Since the IP source address of a packet generally takes no role in forwarding the packet, it can be selected arbitrarily by the sending host without jeopardizing packet delivery. Extra methods are necessary for IP source address validation to augment packet forwarding with an explicit check of whether a given packet's IP source address is legitimate.

IP source address validation can happen at different granularity. Ingress filtering [BCP38] [BCP84], a widely deployed standard for IP source address validation, functions at the coarse granularity of networks. It verifies that the prefix of an IP source address routes to the network from which the packet was received. An advantage of ingress filtering is simplicity: the decision of whether to accept or to reject an IP source address can be made solely based on the information available from routing protocols. However, the simplicity comes at the cost of not being able to validate IP source addresses at a finer granularity, due to the aggregated nature of the information available from routing protocols. Finer-grained IP source address validation would ensure that source address information is accurate, reduce the ability to launch denial-of-service attacks, and help with localizing hosts and identifying misbehaving hosts. Partial solutions [BA2007] exist for finer-grained IP source address validation but are proprietary and hence often unsuitable for corporate procurement.

The Source Address Validation Improvement (SAVI) method was developed to complement ingress filtering with standardized IP source address validation at the maximally fine granularity of individual IP addresses. It prevents hosts attached to the same link from spoofing each other's IP addresses. To facilitate deployment in networks of various kinds, the SAVI method was designed to be modular and extensible. This document describes and motivates the design of the SAVI method.

Note that SAVI raises a number of important privacy considerations that are discussed more fully in [RFC6959]. SAVI implementers must take those privacy considerations into account when designing solutions that match this framework and follow the recommendations given in [RFC6959].

2. Model

To enable network operators to deploy fine-grained IP source address validation without a dependency on supportive functionality on hosts, the SAVI method was designed to be purely network based. A SAVI instance enforces the hosts' use of legitimate IP source addresses according to the following three-step model:

1. Identify which IP source addresses are legitimate for a host, based on monitoring packets exchanged by the host.

2. Bind a legitimate IP address to a link-layer property of the host's network attachment. This property, called a "binding anchor", must be verifiable in every packet that the host sends and harder to spoof than the host's IP source address itself.

3. Enforce that the IP source addresses in packets match the binding anchors to which they were bound.

This model allows SAVI functionality (a SAVI instance) to be located anywhere on the link to which the hosts attach, hence enabling different locations for a SAVI instance. One way to locate a SAVI instance is in the hosts' default router. IP source addresses are then validated in packets traversing the default router, yet the IP source addresses in packets exchanged locally on the link may bypass validation. Another way to locate a SAVI instance is in a switch between the hosts and their default router. Thus, packets may undergo IP source address validation even if exchanged locally on the link.

The closer a SAVI instance is located to the host, the more effective the SAVI method is. This is because each of the three steps of the SAVI model can best be accomplished in a position close to the host:

- Identifying a host's legitimate IP source addresses is most efficient close to the host because the likelihood that the host's packets bypass a SAVI instance, and hence cannot be monitored, increases with the topological distance between the SAVI instance and the host.

- Selecting a binding anchor for a host's IP source address is easiest close to the host because many link-layer properties are unique for a given host only on a link segment directly attached to the host.

- Enforcing a host's use of a legitimate IP source address is most reliable when pursued close to the host because the likelihood that the host's packets bypass a SAVI instance, and hence do not undergo IP source address validation, increases with the topological distance between the SAVI instance and the host.

The preferred location of SAVI instances is therefore close to hosts, such as in switches that directly attach to the hosts whose IP source addresses are being validated.

Nevertheless, it is useful for SAVI mechanisms to be able to handle situations where hosts are not directly attached to the SAVI-capable device. For instance, deployments with both SAVI-capable and legacy switches could be supported. In general, a SAVI solution needs to specify how it deals with a number of deployment scenarios and exceptional situations, including interaction with legacy devices, hosts moving between wireless attachment points, network partitions, and so on.

Besides, in the case of legacy switches, the security level is lower, as there is no full protection for the hosts connected to the legacy server.

3. Deployment Options

The model of the SAVI method, as explained in Section 2, is deployment specific in two ways:

- The identification of legitimate IP source addresses is dependent on the IP address assignment method in use on a link, since it is through assignment that a host becomes the legitimate user of an IP source address.

- Binding anchors are dependent on the technology used to build the link on which they are used, as binding anchors are link-layer properties of a host's network attachment.

To facilitate the deployment of the SAVI method in networks of various kinds, the SAVI method is designed to support different IP address assignment methods and to function with different binding anchors. Naturally, both the IP address assignment methods in use on a link and the available binding anchors have an impact on the functioning and the strength of IP source address validation. The following two subsections explain this impact and describe how the SAVI method accommodates this.

3.1. IP Address Assignment Methods

Since the SAVI method traces IP address assignment packets, it necessarily needs to incorporate logic that is specific to particular IP address assignment methods. However, developing SAVI method variants for each IP address assignment method is alone not sufficient since multiple IP address assignment methods may coexist on a given link. The SAVI method hence comes in multiple variants, e.g., for links with DHCP [RFC2131] [RFC3315], Stateless Address Autoconfiguration (SLAAC) [RFC4862] with or without Secure Neighbor Discovery (SEND) [RFC3971], Internet Key Exchange Protocol Version 2 (IKEv2) [RFC5996] [RFC5739] [RFC5026], and combinations thereof.

The reason to develop SAVI method variants for each single IP address configuration method, in addition to the variant that handles all IP address assignment methods, is to minimize the complexity of the common case. Many link deployments today either are constrained to a single IP address assignment method or, equivalently from the perspective of the SAVI method, use different IP address assignment methods within different IP address prefixes. The SAVI method for such links can be simpler than the SAVI method for links with multiple IP address assignment methods per IP address prefix.

3.2. Binding Anchors

The SAVI method supports a range of binding anchors:

- The IEEE extended unique identifier, EUI-48 or EUI-64, of a host's interface.
- The port on an Ethernet switch to which a host attaches.
- The security association between a host and the base station on wireless links.
- The combination of a host interface's link-layer address and a customer relationship in cable modem networks.
- An ATM virtual channel, a PPP session identifier, or a Layer 2 Tunneling Protocol (L2TP) session identifier in a DSL network.
- A tunnel that connects to a single host, such as an IP-in-IP tunnel, a Generic Routing Encapsulation (GRE) tunnel, or an MPLS label-switched path.

The various binding anchors differ significantly in the security they provide. IEEE extended unique identifiers, for example, fail to render a secure binding anchor because they can be spoofed with little effort. Switch ports alone may be insufficient because they may connect to more than a single host, such as in the case of concatenated switches.

Given this diversity in the security provided, one could define a set of possible binding anchors and leave it up to the administrator to choose one or more of them. Such a selection of binding anchors would, of course, have to be accompanied by an explanation of the pros and cons of the different binding anchors. In addition, SAVI devices may have a default binding anchor depending on the lower layers. Such a default could be to use switch ports when available and MAC addresses otherwise or to use MAC addresses and switch ports in addition if available.

4. Scalability Optimizations

The preference to locate a SAVI instance close to hosts implies that multiple SAVI instances must be able to coexist in order to support large links. Although the model of the SAVI method is independent of the number of SAVI instances per link, coexistence of multiple SAVI instances without further measures can lead to higher-than-necessary memory requirements. Since a SAVI instance creates bindings for the IP source addresses of all hosts on a link, bindings are replicated if multiple SAVI instances coexist on the link. High memory requirements, in turn, increase the cost of a SAVI instance. This is problematic in particular for SAVI instances that are located on a switch since it may significantly increase the cost of such a switch.

To reduce memory requirements for SAVI instances that are located on a switch, the SAVI method enables the suppression of binding replication on links with multiple SAVI instances. This requires manual disabling of IP source address validation on switch ports that connect to other switches running a SAVI instance. Each SAVI instance is then responsible for validating IP source addresses only on those ports to which hosts attach either directly or via switches without a SAVI instance. On ports towards other switches running a SAVI instance, IP source addresses are not validated. The switches running SAVI instances thus form a "protection perimeter". The IP source addresses in packets passing the protection perimeter are validated by the ingress SAVI instance, but no further validation takes place as long as the packets remain within or leave the protection perimeter.

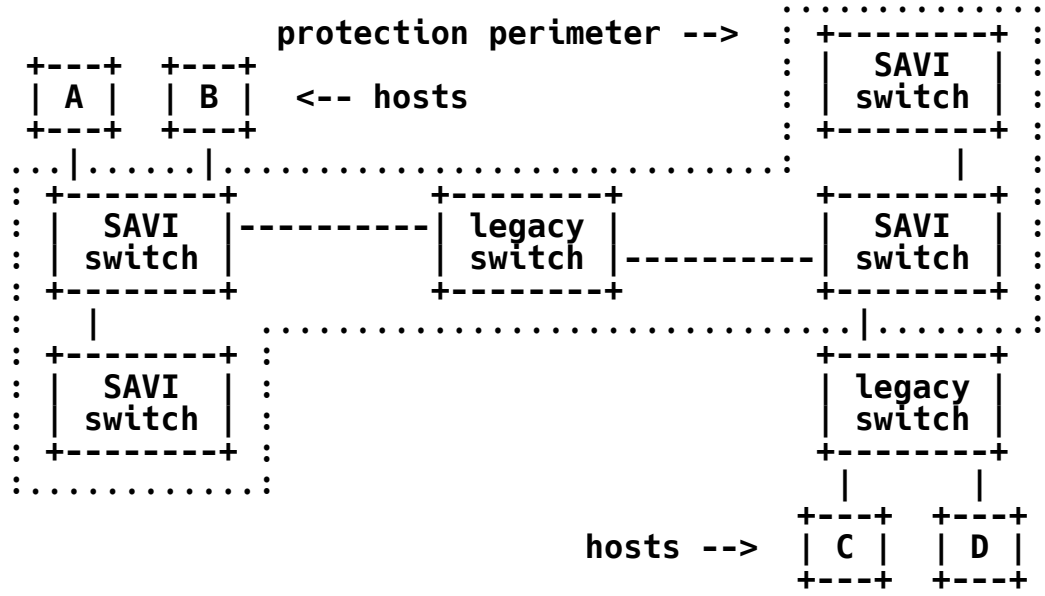

Figure 1: Protection Perimeter Concept

Figure 1 illustrates the concept of the protection perimeter. The figure shows a link with six switches, of which four, denoted "SAVI switch", run a SAVI instance. The protection perimeter created by the four SAVI instances is shown as a dotted line in the figure. IP source address validation is enabled on all switch ports on the protection perimeter, and it is disabled on all other switch ports. Four hosts, denoted A through D in the figure, attach to the protection perimeter.

In the example in Figure 1, the protection perimeter encompasses one of the legacy switches, located in the middle of the depicted link topology. This enables a single, unpartitioned protection perimeter. A single protection perimeter minimizes memory requirements for the SAVI instances because every binding is kept only once, namely, by the SAVI instance that attaches to the host being validated. Excluding the legacy switch from the protection perimeter would result in two smaller protection perimeters to the left and to the right of the depicted link topology. The memory requirements for the SAVI instances would then be higher: since IP source address validation would be activated on the two ports connecting to the legacy switch, the SAVI instances adjacent to the legacy switch would replicate all bindings from the other protection perimeter, respectively. The reason why it is possible to include the legacy switch in the protection perimeter is because the depicted link topology guarantees that packets cannot enter the protection perimeter via this legacy switch. Without this guarantee, the legacy

switch would have to be excluded from the protection perimeter in order to ensure that packets entering the protection perimeter undergo IP source address validation.

Note that if such configuration is used, care must be taken as any hosts on subnets attached to non-enforcing ports will be able to use spoofed source addresses.

5. Reliability Optimizations

The explicit storage of legitimate IP addresses in the form of bindings implies that failure to create a binding, or the premature removal of bindings, can lead to loss of legitimate packets. There are three situations in which this can happen:

- Legitimate IP address configuration packets, which should trigger the creation of a binding in a SAVI instance, are lost before reaching the SAVI instance.
- A SAVI instance loses a binding, for example, due to a restart.
- The link topology changes, resulting in hosts to communicate through SAVI instances that do not have a binding for those hosts' IP addresses.

To limit the disruption that missing bindings for legitimate IP addresses can have, the SAVI method includes a mechanism for reactive binding creation based on regular packets. This mechanism supplements the proactive binding creation based on IP address configuration packets. Reactive binding creation occurs when a SAVI instance recognizes excessive drops of regular packets originating from the same IP address. The SAVI instance then verifies whether said IP address is unique on the link. How the verification is carried out depends on the IP address configuration method that the SAVI instance supports. The SAVI method variant for Stateless Address Autoconfiguration and for Secure Neighbor Discovery verifies an IP address through the Duplicate Address Detection procedure. The SAVI method variant for DHCP verifies an IP address through a DHCP Lease Query message exchange with the DHCP server. If verification indicates that the IP address is unique on the link, the SAVI instance creates a binding for the IP address. Otherwise, no binding is created, and packets sent from the IP address continue to be dropped. These reliability issues should be addressed in all the SAVI protocols describing particular SAVI methods.

6. Scenario with Multiple Assignment Methods

While multiple assignment methods can be used on the same link, the SAVI device may have to deal with a mix of binding discovery methods. If the address prefix used for each assignment method is different, the "mix scenario" behaves the same as with the scenario with only one assignment method. If different address assignment methods are used to assign addresses from the same prefix, additional considerations are needed because one binding mechanism may create a binding violating an existing binding from another binding mechanism, e.g., binding from First-Come, First-Served (FCFS) SAVI [RFC6620] may violate a binding from SAVI-DHCP [SAVI-DHCP]. Thus, the collision between different SAVI mechanisms in the mix scenario must be handled in case more than one address assignment method is used to assign addresses from the same prefix.

The prioritization of relationships between different address assignment methods is used as the basis to solve possible collisions. Current standard documents of address assignment methods (DHCP [RFC2131], DHCPv6 [RFC3315], SLAAC [RFC4862], and SEND [RFC3971]) have implied the prioritization relationship in general cases. However, in some scenarios, the default prioritization level may not be quite suitable. A configurable prioritization level should be supported in the SAVI solution for the mix scenario [SAVI-MIX].

7. Prefix Configuration

Before setting up a host-level granularity binding, it is important to configure correct prefixes on the SAVI device. This document suggests a set of 3 prefix configuration mechanisms at a SAVI device:

- Manual Prefix Configuration: The allowed prefix scope of IPv4 addresses, IPv6 static addresses, IPv6 addresses assigned by Stateless Address Autoconfiguration (SLAAC), and IPv6 addresses assigned by DHCPv6 can be set manually at the SAVI device. FE80::/64 is always a feasible prefix in IPv6.

- Prefix Configuration by Router Advertisement (RA) Snooping: The allowed prefix scope of IPv6 static addresses and IPv6 addresses assigned by SLAAC can be set at the SAVI device through snooping an RA message at the SAVI device.

- Prefix Configuration by DHCP Prefix Delegation (DHCP-PD) Snooping: The allowed prefix scope of IPv6 static addresses, IPv6 addresses assigned by SLAAC, and IPv6 addresses assigned by DHCPv6 can be set through snooping a DHCP-PD message at the SAVI device.

If some of the prefix scopes are set to have no prefix, the implication is that the corresponding address assignment method is not allowed in the network.

There is no need to explicitly present these prefix scopes, but these restrictions should be used as the premier check in binding setup.

When SAVI is partially deployed, binding may fail as RA messages and DHCP-PD can be spoofed. So, it is recommended that Manual Prefix Configuration be used unless SAVI gets fully deployed.

8. Acknowledgments

The authors would like to thank the SAVI working group for a thorough technical discussion on the design and the framework of the SAVI method as captured in this document, in particular Erik Nordmark, Guang Yao, Eric Levy-Abegnoli, and Alberto Garcia. Thanks also to Torben Melsen for reviewing this document.

9. Security Considerations

This document only discusses the possible methods to mitigate the usage of forged IP addresses. Some such methods may rely on cryptographic methods, but not all do. As a result, it is generally not possible to prove address ownership in any strong sense. If a binding anchor is not exclusive for each IP address, or is without strong security, addresses can still be forged. Besides, the binding may not accord with the address management requirement, which can be more specified for each client. However, given no new protocol is introduced, the improvements are still acceptable. If strong security is required when using IP addresses, then cryptographic-based authentication must be used as it is the only way to provide strong security.

Section 2 explains how the preferred location of SAVI instances is close to hosts. However, in some cases, this makes the SAVI instances themselves vulnerable and may defeat the purpose of deploying a SAVI solution. For instance, deployments should not place SAVI functionality in devices that are physically exposed. Even if the device correctly monitors the source address usage of hosts, an attacker could replace the device with one that does not check or hook up to a trusted interface from the device to the rest of the network. Similarly, deployments where SAVI instances are distributed across administrative boundaries are not recommended. For instance, in most cases, it would be undesirable to deploy a distributed SAVI solution on both sides of a customer-provider interface if the solution required trusting the correct operation of the SAVI devices on the other side of the interface.

10. References

10.1. Normative References

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC5026] Giaretta, G., Kempf, J., and V. Devarapalli, "Mobile IPv6 Bootstrapping in Split Scenario", RFC 5026, October 2007.

[RFC5739] Eronen, P., Laganier, J., and C. Madson, "IPv6 Configuration in Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5739, February 2010.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

[RFC6620] Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses", RFC 6620, May 2012.

[RFC6959] McPherson, D., Baker, F., and J. Halpern, "Source Address Validation Improvement (SAVI) Threat Scope", RFC 6959, May 2013.

10.2. Informative References

[BA2007] Baker, F., "Cisco IP Version 4 Source Guard", Work in Progress, November 2007.

[BCP38] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[BCP84] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.

[SAVI-DHCP] Bi, J., Wu, J., Yao, G., and F. Baker, "SAVI Solution for DHCP", Work in Progress, June 2013.

[SAVI-MIX] Bi, J., Yao, G., Halpern, J., and E. Levy-Abegnoli, "SAVI for Mixed Address Assignment Methods Scenario", Work in Progress, May 2013.

Authors' Addresses

Jianping Wu
Tsinghua University
Computer Science, Tsinghua University
Beijing 100084
China

EMail: jianping@cernet.edu.cn

Jun Bi
Tsinghua University
Network Research Center, Tsinghua University
Beijing 100084
China

EMail: junbi@tsinghua.edu.cn

Marcelo Bagnulo
Universidad Carlos III de Madrid
Avenida de la Universidad 30
Leganes, Madrid 28911
Spain

EMail: marcelo@it.uc3m.es

Fred Baker
Cisco Systems
Santa Barbara, CA 93117
United States

EMail: fred@cisco.com

Christian Vogt (editor)
3507 Palmilla Drive
San Jose, CA 95134
United States

EMail: mail@christianvogt.net