Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 26, 2011

Jianping Wu
Jun Bi
CERNET
Marcelo Bagnulo
UC3M
Fred Baker
Cisco
Christian Vogt, Ed.
Ericsson

September 22, 2010

Source Address Validation Improvement Protocol Framework
draft-ietf-savi-framework-00

Abstract

The Source Address Validation Improvement protocol was developed to complement ingress filtering with finer-grained, standardized IP source address validation. This document describes and motivates the design of the SAVI protocol.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on March 26, 2011.

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. 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], 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 be helpful to enable IP-source-address-based authentication, authorization, and host localization, as well as to efficiently identify 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 protocol 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 protocol was designed to be modular and extensible. This document describes and motivates the design of the SAVI protocol.

2. Protocol Model

To enable network operators to deploy fine-grained IP source address validation without a dependency on supportive functionality on hosts, the SAVI protocol was designed to be purely network-based. A SAVI protocol instance is located on the path of hosts' packets, enforcing

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 a SAVI protocol instance to be located anywhere on the link to which the hosts attach, hence enabling different locations for a protocol instance. One way to locate a SAVI protocol 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 protocol 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 protocol instance is located to the hosts, the more effective the SAVI protocol is. This is because each of the three steps of the SAVI protocol 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 protocol instance, and hence cannot be monitored, increases with the distance between the SAVI protocol 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 attaching 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 protocol instance, and hence do not undergo IP source address validation, increases with the distance between the protocol instance and the host.

The preferred location of SAVI protocol instances is therefore close

to hosts, such as in switches that directly attach to the hosts whose IP source addresses are being validated.

3. Deployment Options

The model of the SAVI protocol, 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 protocol in networks of various kinds, the SAVI protocol 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 sub-sections explain this impact, and describe how the SAVI protocol accommodates this.

3.1. IP Address Assignment Methods

Since the SAVI protocol traces IP address assignment packets, it necessarily needs to incorporate logic that is specific to particular IP address assignment methods. However, developing SAVI protocol variants for each IP address assignment method is alone not sufficient, since multiple IP address assignment methods may co-exist on a given link. The SAVI protocol hence comes in multiple variants: for links with Stateless Address Autoconfiguration, for links with DHCP, for links with Secure Neighbor Discovery, and for links that use any combination of IP address assignment methods.

The reason to develop SAVI protocol 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 methods or, equivalently from the perspective of the SAVI protocol, separate IP address assignment methods into different IP address prefixes. The SAVI protocol for such links can be simpler than the SAVI protocol for links with multiple IP address assignment methods per IP address prefix.

3.2. Binding Anchors

The SAVI protocol 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 PPPoE session identifier, or an L2TP session identifier in a DSL network.
- A tunnel that connects to a single host, such as an IP-in-IP tunnel, a 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. And 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 protocol instance close to hosts implies that multiple SAVI protocol instances must be able to co-exist in order to support large links. Although the SAVI protocol model is independent of the number of protocol instances per link, co-existence of multiple protocol instances without further measures can lead to higher-than-necessary memory requirements: Since a SAVI protocol instance creates bindings for the IP source addresses of all

hosts on a link, bindings are replicated if multiple protocol instances co-exist on the link. High memory requirements, in turn, increase the cost of a SAVI protocol instance. This is problematic in particular for SAVI protocol instances that are located on a switch, since it may significantly increase the cost of such a switch.

To reduce memory requirements for SAVI protocol instances that are located on a switch, the SAVI protocol enables the suppression of binding replication on links with multiple protocol instances. This requires manual disabling of IP source address validation on switch ports that connect to other switches running a SAVI protocol instance. Each SAVI protocol 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 protocol instance. On ports towards other switches running a SAVI protocol instance, IP source addresses are not validated. The switches running SAVI protocol instances thus form a "protection perimeter". The IP source addresses in packets passing the protection perimeter are validated by the ingress SAVI protocol instance, but no further validation takes place as long as the packets remain within, or leave the protection perimeter.

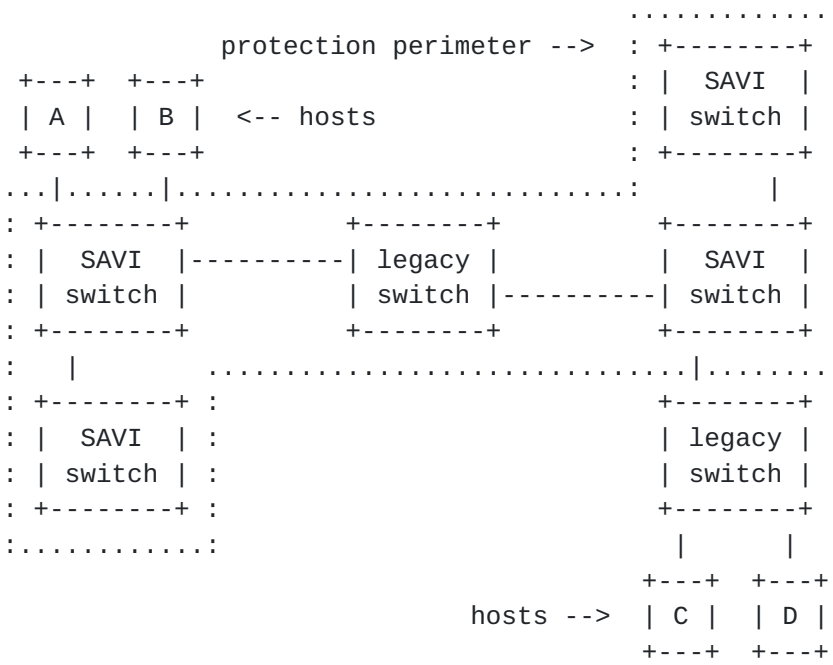

Figure 1: Protection perimeter concept

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 protocol instance. The protection perimeter created by the four SAVI protocol 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 of figure 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 protocol instances because every binding is kept only once, namely, by the SAVI protocol 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 protocol instances would then be higher: Since IP source address validation would be activated on the two ports connecting to the legacy switch, the SAVI protocol instances adjacent to the legacy switch would replicate all bindings from the respectively other protection perimeter. 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.

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 protocol instance, are lost before reaching the SAVI protocol instance.

- A SAVI protocol instance loses a binding, for example, due to a restart.

- The link topology changes, resulting in hosts to communicate through SAVI protocol 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 protocol 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 protocol instances recognizes excessive drops of regular packets originating from the same IP address. The SAVI protocol 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 protocol instance supports: The SAVI protocol variant for Stateless Address Autoconfiguration and for Secure Neighbor Discovery verifies an IP address through the Duplicate Address Detection procedure. The SAVI protocol 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 protocol instance creates a binding for the IP address. Otherwise, no binding is created, and packets sent from the IP address continue to be dropped.

6. Acknowledgment

The author would like to thank the SAVI working group for a thorough technical discussion on the design and the framework of the SAVI protocol, 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.

This document was generated using the xml2rfc tool.

7. References

[BA2007] Fred, F., "Cisco IP Version 4 Source Guard", IETF Internet draft (work in progress), November 2007.

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

Authors' Addresses

Jianping Wu
CERNET
Computer Science, Tsinghua University
Beijing 100084
China

Email: jianping@cernet.edu.cn

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

Email: junbi@cernet.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)
Ericsson
200 Holger Way
San Jose, CA 95134
United States

Email: christian.vogt@ericsson.com