SAVI
Internet-Draft
Intended status: Standards Track
Expires: November 17, 2014

J. Bi
G. Yao
Tsinghua Univ.
J. Halpern
Newbridge
E. Levy-Abegnoli, Ed.
Cisco

May 16, 2014

SAVI for Mixed Address Assignment Methods Scenario
draft-ietf-savi-mix-06

Abstract

This document reviews how multiple address discovery methods can coexist in a single SAVI device and collisions are resolved when the same binding entry is discovered by two or more methods.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on November 17, 2014.

Table of Contents

1. Introduction

There are currently several documents [savi-fcfs], [savi-dhcp] and [savi-send] that describe the different methods by which a switch can discover and record bindings between a node's layer3 address and a binding anchor and use that binding to perform Source Address Validation. Each of these documents specifies how to learn on-link addresses, based on the method used for their assignment, respectively: StateLess Autoconfiguration (SLAAC), Dynamic Host Control Protocol (DHCP) and Secure Neighbor Discovery (SeND). Each of these documents describes separately how one particular discovery method deals with address collisions (same address, different anchor).

While multiple assignment methods can be used in the same layer2 domain, a SAVI device might have to deal with a mix of binding discovery methods. The purpose of this document is to provide recommendations to avoid collisions and to review collisions handling when two or more such methods come up with competing bindings.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [rfc2119].

3. Problem Scope

There are three address assignment methods identified and reviewed in one of the SAVI document:

1. StateLess Address AutoConfiguration (SLAAC) - reviewed in [savi-fcfs]

2. Dynamic Host Control Protocol address assignment (DHCP) - reviewed in [savi-dhcp]

3. Secure Neighbor Discovery (SeND) address assignment, reviewed in [savi-send]

Each address assignment method corresponds to a binding discovery method: SAVI-FCFS, SAVI-DHCP and SAVI-SeND. In addition, there is a fourth method for installing a bindings on the switch, referred to as "manual". It is based on manual (address or prefix) binding configuration and is reviewed in [savi-fcfs] and [savi-framework].

All combinations of address assignment methods can coexist within a layer2 domain. A SAVI device will have to implement the corresponding SAVI discovery methods (referred to as a "SAVI solution") to enable Source Address Validation. If more than one SAVI solution is enabled on a SAVI device, the method is referred to as "mix address assignment method" in this document.

SAVI solutions are independent from each other, each one handling its own entries. In the absence of reconciliation, each solution will reject packets sourced with an address it did not discovered. To prevent addresses discovered by one solution to be filtered out by another, the binding table should be shared by all the solutions. However this could create some conflict when the same entry is discovered by two different methods: the purpose of this document is of two folds: provide recommendations and method to avoid conflicts, and resolve conflicts if and when they happen. Collisions happening within a given solution are outside the scope of this document.

4. Architecture

A SAVI device may enable multiple SAVI methods. This mechanism, called SAVI-MIX, is proposed as a layer between the binding generation algorithems and the binding database which contains the working binding entries Figure 1. SAVI methods, i.e., SAVI-FCFS, SAVI-DHCP, SAVI-SEND, do not have exclusive binding tables. Once a SAVI method generates a candidate binding, it will request SAVI-MIX to set up a corresponding entry in the shared binding database, named Binding DB. Then SAVI-MIX will check if there is any conflict in the Binding DB. A new binding will be generated if there is no conflict. If there is a conflict, SAVI-MIX will determine whether replace the existing binding or reject the candidate binding based on the policies specified in Section 7. Whether the candidate binding can be install in the Binding DB will not be returned to the requesting SAVI method.

Correspondingly, the packet filtering will not be performed by each SAVI method separately. Instead, SAVI-MIX will perform filtering based on the entries in the Binding DB.

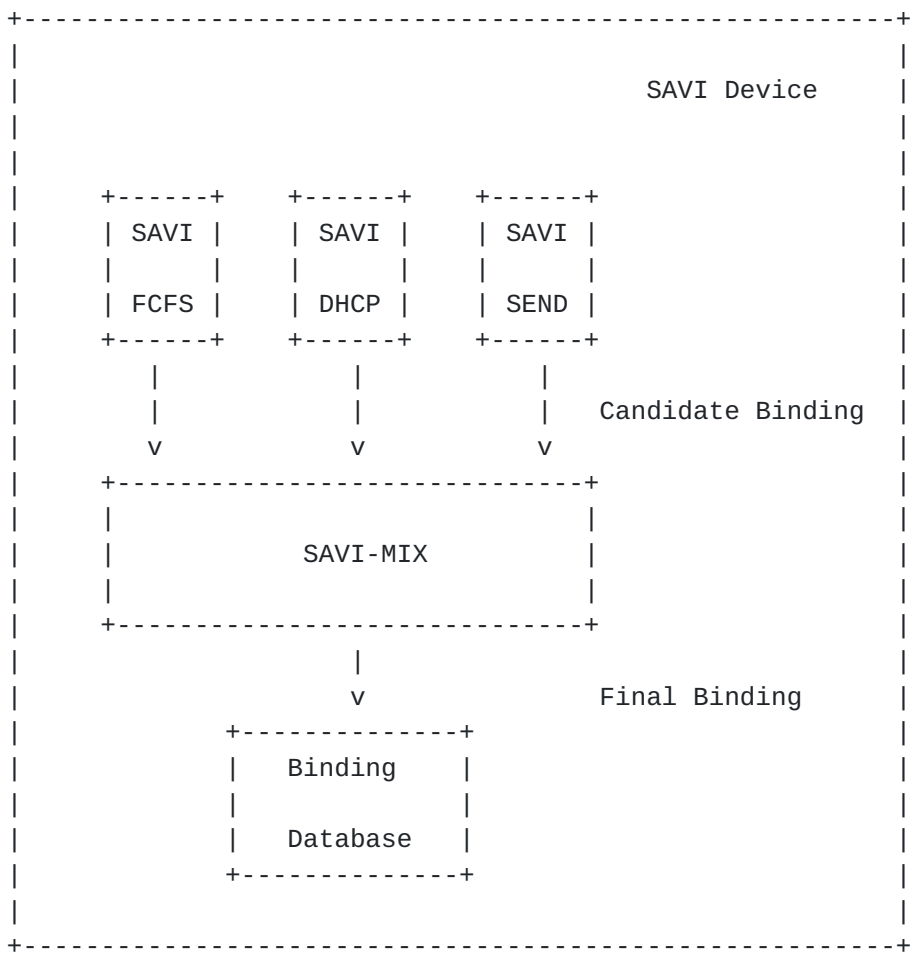

Figure 1: SAVI-Mix Architecture

5. Security Perimeter in SAVI MIX

The perimeter of SAVI MIX is the union of the perimeter of each SAVI method, as illustrated in Figure 2.

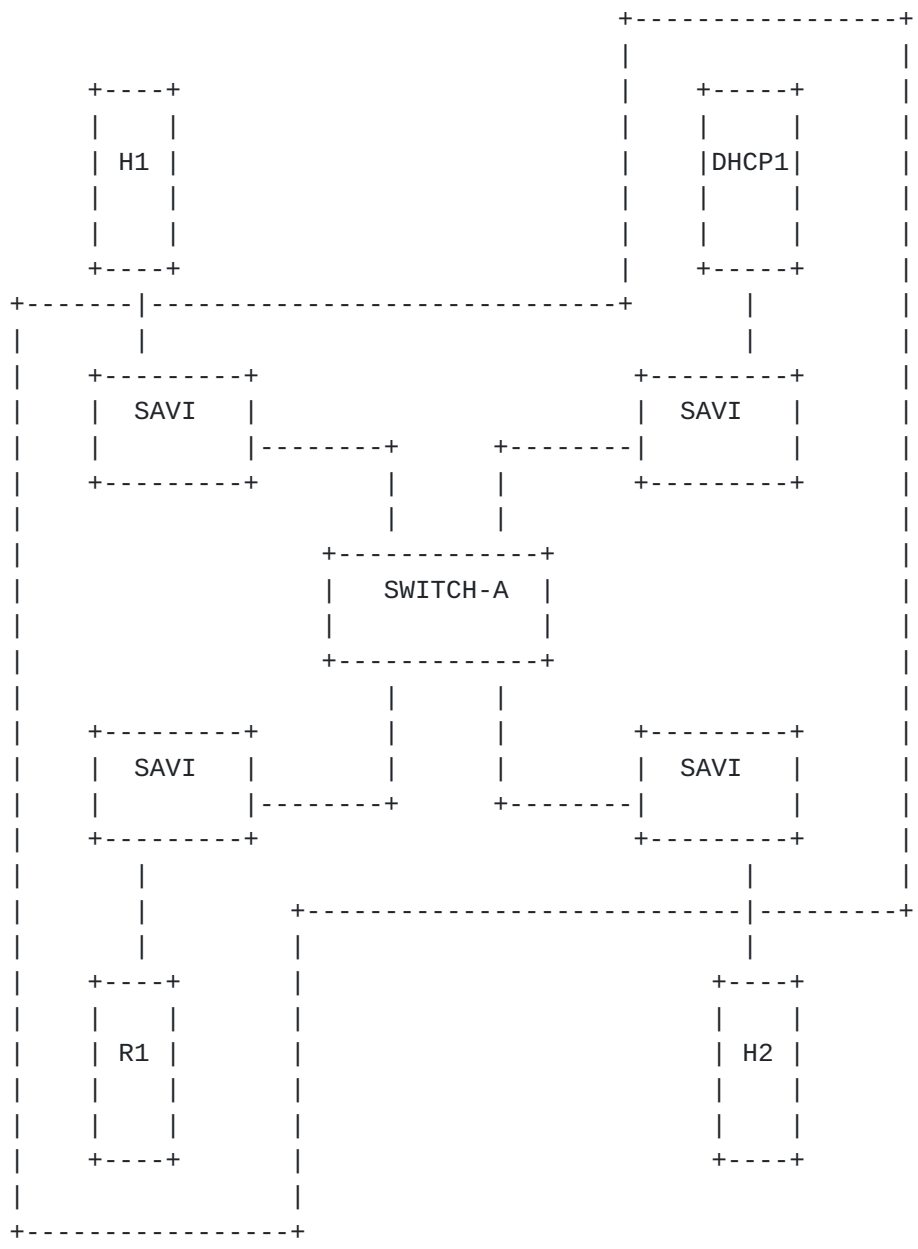

Figure 2: SAVI-Mix Perimeter

6. Recommendations for preventing collisions

If each solution has a dedicated address space, collisions won't happen. Using non overlapping address space across SAVI solutions is therefore recommended. To that end, one should:

1. DHCP/SLAAC: use non-overlapping prefix for DHCP and SLAAC. Set the A bit in Prefix information option of Router Advertisement for SLAAC prefix. And set the M bit in Router Advertisement for DHCP prefix. For detail explanations on these bits, refer to [rfc4861][rfc4862].

2. SeND/non-SeND: avoid mixed environment (where SeND and non-SeND nodes are deployed) or separate the prefixes announced to SeND and non-SenD nodes. One way to separate the prefixes is to have the router(s) announcing different (non-overlapping) prefixes to SeND and to non-SeND nodes, using unicast Router Advertisements, in response to SeND/non-SeND Router Solicit.

7. Handing binding collisions

In situations where collisions could not be avoided, two cases should be considered:

1. The same address is bound on two different binding anchors by different SAVI solutions.

2. The same address is bound on the same binding anchor by different SAVI solutions.

7.1. Same Address on Different Binding Anchors

This would typically occur in case assignment address spaces could not be separated. For instance,overl an address is assigned by SLAAC on node X, installed in the binding table using SAVI-FCFS, anchored to "anchor-X". Later, the same address is assigned by DHCP to node Y, as a potential candidate in the same binding table, anchored to "anchor-Y".

7.1.1. Basic preference

The SAVI device must decide whom the address should be bound with (anchor-X or anchor-Y in this example). Current standard documents of address assignment methods have implied the prioritization relationship (first-come). In the absence of any configuration or protocol hint (see Section 7.1.2) the SAVI device should choose the first-come entry, whether it was learnt from SLACC, SeND or DHCP.

7.1.2. Overwritten preference

There are two identified exceptions to the general prioritization model, one of them being CGA addresses, another one controlled by the configuration of the switch:

1. When CGA addresses are used, and a collision is detected, preference should be given to the anchor that carries the CGA credentials once they are verified, in particular the CGA parameters and the RSA options. Note that if an attacker was trying to replay CGA credentials, he would then compete on the base of fcfs (first-come, first-serve).

2. The SAVI device should allow the configuration of a triplet ("prefix", "anchor", "method") or ("address", "anchor", "method"). Later, if a DAD message is received for a target within "prefix" (or equal "address") bound to "anchor1" (different from "anchor"), or via a discovery method different from "method", the switch should defend the address by responding to the DAD message. It should not at this point install the entry into the binding table. It will simply prevent the node to assign the address, and will de-facto prioritize the configured anchor or configured assignment method for that address. This is especially useful to protect well known bindings such as a static address of a server over anybody, even when the server is down. It is also a way to give priority to a binding learnt from SAVI-DHCP over a binding for the same address, learnt from SAVI-FCFS.

7.1.3. Multiple SAVI Device Scenario

A single SAVI device doesn't have the information of all bound addresses on the perimeter. Therefore it is not enough to lookup local bindings to identify a collision. However, assuming DAD is performed throughout the security perimeter for all addresses regardless of the assignment method, then DAD response will inform all SAVI devices about any collision. In that case, FCFS will apply the same way as in a single switch scenario. If the admin configured on one the switches a prefix (or a single static binding) to defend, the DAD response generated by this switch will also prevent the binding to be installed on other switches of the perimeter.

7.2. Same Address on the Same Binding Anchor

A binding may be set up on the same binding anchor by multiple solutions. For example, if SAVI-FCFS and SAVI-DHCP are both enabled on one SAVI device, a DHCP address be bound by both SAVI instances.

There is no conflict if the binding is valid in all the solutions. However, the binding lifetimes of different solutions can be different. If one SAVI instance changes the state of a binding to invalid on lifetime expires, conflict will happen.

The solution proposed is to keep a binding as long as possible. A binding is kept until it has been required to be removed by all the

solutions that ever set up it.

8. Security Considerations

As described in [savi-framework], this solution cannot strictly prevent spoofing. There are two scenarios in which spoofing can still happen:

1. The binding anchor is spoofable. if the binding anchor is spoofable, e.g., plain MAC address, an attacker can use forged binding anchor to send packet which will not be regarded as spoofing by SAVI device. Indeed, using binding anchor that can be easily spoofed is dangerous. An attacker can use the binding anchor of another host to perform a lot of DHCP procedures, and the SAVI device will refuse to set up new binding for the host whenever the binding number limitation has been reached. Thus, it is RECOMMENDED to use strong enough binding anchor, e.g., switch port, secure association in 802.11ae/af and 802.11i.

2. The binding anchor is shared by more than one host. If the binding anchor is shared by more than one host, they can spoof the addresses of each other. For example, a number of hosts can attach to the same switch port of a SAVI device through a hub. The SAVI device cannot distinguish packets from different hosts and thus the spoofing between them will not be detected. This problem can be solved through not sharing binding anchor between hosts.

9. IANA Considerations

This memo asks the IANA for no new parameters.

Note to RFC Editor: This section will have served its purpose if it correctly tells IANA that no new assignments or registries are required, or if those assignments or registries are created during the RFC publication process. From the authors' perspective, it may therefore be removed upon publication as an RFC at the RFC Editor's discretion.

10. Acknowledgment

Thanks to Christian Vogt, Eric Nordmark, Marcelo Bagnulo Braun and Jari Arkko for their valuable contributions.

This document was generated using the xml2rfc tool.

11. References

11.1. Informative References

[rfc2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, Match 1997.

11.2. Normative References

[rfc4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[rfc4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[savi-dhcp] Bi, J., Wu, J., Yao, G., and F. Baker, "SAVI Solution for DHCP", draft-ietf-savi-dhcp-18 (work in progress), February 2012.

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

[savi-framework] Wu, J., Bi, J., Bagnulo, M., Baker, F., and C. Vogt, Ed., "Source Address Validation Improvement Framework", RFC 7039, October 2013.

[savi-send] Bagnulo, M. and A. Garcia-Martinez, "SEND-based Source-Address Validation Implementation", draft-ietf-savi-send-06 (work in progress), October 2011.

Authors' Addresses

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

Email: junbi@tsinghua.edu.cn

Guang Yao
Tsinghua University
Network Research Center, Tsinghua University
Beijing 100084
China

Email: yaoguang@cernet.edu.cn

Joel M. Halpern
Newbridge Networks Inc

Email: jmh@joelhalpern.com

Eric Levy-Abegnoli (editor)
Cisco Systems
Village d'Entreprises Green Side - 400, Avenue Roumanille
Biot-Sophia Antipolis 06410
France

Email: elevyabe@cisco.com