Network Working Group   J. Bi
Internet-Draft   CERNET
Intended status: Standards Track   G. Yao
Expires: September 15, 2011   Tsinghua University
                                                          J. Halpern
                                                Newbridge Networks Inc
                                             E. Levy-Abegnoli, Ed.
                                                        Cisco Systems
                                                       March 14, 2011

                SAVI for Mixed Address Assignment Methods Scenario
                     <draft-bi-savi-mix-04.txt>

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

Copyright Notice

Table of Contents

1. Introduction

   There are currently several documents [I-D.ietf-savi-fcfs],
   [I-D.ietf-savi-dhcp], [I-D.ietf-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.

   The method used by nodes to assign the address drove the break down
   into these multiple documents, whether StateLess Autoconfiguration
   (SLAAC), Dynamic Host Control Protocol (DHCP), Secure Neighbor
   Discovery (SeND) or manual. Each of these documents describes
   separately how one particular discovery method deals with address
   collisions.

   While multiple assignment methods can be used in the same layer2
   domain, a savi-switch 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. Mixed Address Assignment Methods Scenario

   There are four address assignment methods identified and reviewed in
   one of the SAVI document:
   1. StateLess Address AutoConfiguration (SLAAC) - reviewed in
       [I-D.ietf-savi-fcfs]
   2. Dynamic Host Control Protocol address assignment (DHCP) -
       reviewed in [I-D.ietf-savi-dhcp]
   3. Secure Neighbor Discovery (SeND) address assignment, reviewed in
       [I-D.ietf-savi-send]
   4. Manually address configuration - reviewed in [I-D.ietf-savi-fcfs]
       and [I-D.ietf-savi-framework]

   Each address assignment method corresponds to a binding discovery
   method: SAVI-FCFS, SAVI-DHCP and SAVI-SeND.

   Any combination of address assignment methods can be potentially
   mixed within a layer2 domain, and a savi device will have to
   implement the corresponding savi discovery method (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.

3. Problem Scope, Statement and Solution

3.1. Problem Scope

   Different savi solutions are independent from each other, each one
   handling its own entries. In the absence of a 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 to avoid conflicts,
   and resolve conflicts if and when they happen. Collisions happening
   within a given solution is outside the scope of this document.

3.2. Recommendations for preventing collisions

   If each solution has a dedicated address space, collisions won't
   happen. Thus, it is recommended to avoid overlap in the address
   space across SAVI solutions enabled on any particular savi switch.
   More specifically:

   1. DHCP/Static: exclude the static address from the DHCP pool.
   2. DHCP/SLAAC: separate the prefix scope of 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. [RFC4861] [RFC4862].
   3. SLAAC/Static: separate the prefix scope of SLAAC and Static. It
       may be impossible in practice. SAVI device can perform DAD proxy
       for static address to hold the address from SLAAC node.
   4. SEND/non-SEND: In an environment where SeND is deployed, the only
       way to avoid collisions in the SAVI devices is to have SeND-only
       nodes. In a mixed environment, two nodes, SeND and non-SeND,
       could configure the same address and the SAVI-device will have to
       deal with a collision.

3.3. Binding on the Same Address

   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.

3.3.1. Same Address on Different Binding Anchors

   This is the very case of collision that could not be prevented by
   separating the assignment address spaces. For instance, 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".

3.3.1.1. Basic preference

   Within the SAVI perimeter, one address bound to a binding anchor by
   one SAVI solution could also be bound by another SAVI solution to a
   different binding anchor. If the DAD procedure is not performed, the
   same address will also be bound to the new binding anchor. Both
   bindings are legitimate within the corresponding solution.

   Though it is possible that the hosts and network can still work in
   such scenario, the uniqueness of address is not assured. The SAVI
   device must decide whom the address should be bound with. A binding
   preference level based solution is proposed here.

   To determine a proper preference level, following evidences are used:
   1. "Duplicate Address Detection MUST be performed on all unicast
       addresses prior to assigning them to an interface, regardless of
       whether they are obtained through stateless autoconfiguration,
       DHCPv6, or manual configuration,..." [RFC4862]
   2. "A tentative address that is determined to be a duplicate as
       described above MUST NOT be assigned to an interface,..."
       [RFC4862]
   3. "The client SHOULD perform duplicate address detection on each of
       the addresses in any IAs it receives in the Reply message before
       using that address for traffic." [RFC3315]
   4. A SEND node that uses the CGA authorization method to protect
       Neighbor Solicitations SHOULD perform Duplicate Address Detection
       as follows. If Duplicate Address Detection indicates that the
       tentative address is already in use, the node generates a new
       tentative CGA. If after three consecutive attempts no non-unique
       address is generated, it logs a system error and gives up
       attempting to generate an address for that interface.

       When performing Duplicate Address Detection for the first
       tentative address, the node accepts both secured and unsecured
       Neighbor Advertisements and Solicitations received in response to
       the Neighbor Solicitations. When performing Duplicate Address
       Detection for the second or third tentative address, it ignores
       unsecured Neighbor Advertisements and Solicitations." [RFC3971]

5. "The node MAY have a configuration option whereby it ignores
    unsecured advertisements, even when performing Duplicate Address
    Detection for the first tentative address. This configuration
    option SHOULD be disabled by default. This is a recovery
    mechanism for cases in which attacks against the first address
    become common." [RFC3971]

From the above materials, "First-Come First-Serve" should be the
default behavior for choosing between two competing bindings. There
can however be some exceptions, 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.

2. The switch configuration should allow an address range (including
    a single address) to be configured together with a given anchor
    or constrained to be discovered by a particular savi-solution.
    If a DAD message for a target within that range is received on
    the savi-switch from an anchor, or via a discovery method
    different from the one configured, the switch should defend the
    address by responding to the DAD message. 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.

    Note that no binding shall be created in the binding table until
    an "acceptable" address owner shows up, either from the
    configured anchor or using the savi solution associated with that
    address.

The following preference level can be inferred from listed materials
and above analysis:
1. By default, SLAAC, DHCP and manually configured address by user
    have the same priority.
2. SEND can have higher priority because it may configure an address
    bound by non-SEND node.
3. Static binding configured on the switch (admin) will have the
    highest priority
4. Address range configured on the switch (admin) constrained to
    DHCP discovery will de-facto be given a higher priority over
    FCFS, by defending the address until it is is effectively learnt
    from DHCP

Combined solution preference with binding sequence, there will be 16

scenarios (Denote solutions by FCFS, DHCP, SEND, and Admin
correspondingly):

Existing	Candidate	Default PREFERENCE
FCFS	FCFS	In the scope of SAVI-SLAAC
FCFS	DHCP	FCFS
FCFS	SEND	SEND
FCFS	Admin	Admin
DHCP	FCFS	DHCP
DHCP	DHCP	In the scope of SAVI-DHCP
DHCP	SEND	SEND
DHCP	Admin	Admin
SEND	FCFS	SEND
SEND	DHCP	SEND
SEND	SEND	In the scope of SAVI-SEND
SEND	Admin	Admin
Admin	FCFS	Admin
Admin	DHCP	Admin
Admin	SEND	Admin
Admin	Admin	Candidate binding

3.3.1.2. 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 switches 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 range of addresses (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.

3.3.1.3. Conflict Announcement

If a host is prohibited from using a bound address, the violation
MUST be announced to it, through delivering one (or more) Neighbor
Advertisement message to the host.

3.3.2. Same Address on the Same Binding Anchor

A binding may be set up on the same binding anchor by multiple
solutions. Generally, the binding lifetimes of different solutions
are different. Potentially, if one solution requires to remove the
binding, the node using the address may be taken the use right.

For example, a node performs DAD procedure after being assigned an
address from DHCP, then the address will also be bound by SAVI-FCFS.
If the SAVI-FCFS lifetime is shorter than DHCP lifetime, when the
SAVI-FCFS lifetime expires, it will request to remove the binding.
If the binding is removed, the node will not be able to use the
address even the DHCP lease time doesn't expire.

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.

4. References

4.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", BCP 14, RFC 2119, March 1997.

4.2. Informative References

[I-D.ietf-savi-dhcp]
           Bi, J., Wu, J., Yao, G., and F. Baker, "SAVI Solution for
           DHCP", draft-ietf-savi-dhcp-07 (work in progress),
           November 2010.

[I-D.ietf-savi-fcfs]
           Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS-
           SAVI: First-Come First-Serve Source-Address Validation for
           Locally Assigned Addresses", draft-ietf-savi-fcfs-05 (work
           in progress), October 2010.

[I-D.ietf-savi-framework]
           Wu, J., Bi, J., Bagnulo, M., Baker, F., and C. Vogt,
           "Source Address Validation Improvement Framework",
           draft-ietf-savi-framework-03 (work in progress),
           March 2011.

[I-D.ietf-savi-send]
           Bagnulo, M. and A. Garcia-Martinez, "SEND-based Source-
           Address Validation Implementation",
           draft-ietf-savi-send-04 (work in progress), October 2010.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6
           (IPv6) Specification", RFC 2460, December 1998.

[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.

   [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)",
                    RFC 3972, March 2005.

   [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.

Appendix A. Contributors and Acknowledgments

Authors' Addresses

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

   Email: junbi@cernet.edu.cn

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

   Email: yaoguang.china@gmail.com

   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