SAVI Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 6, 2012

M. Bagnulo
A. Garcia-Martinez
UC3M
October 4, 2011

SEND-based Source-Address Validation Implementation
draft-ietf-savi-send-06

Abstract

This memo describes SEND SAVI, a mechanism to provide source address validation using the SEND protocol. The proposed mechanism is intended to complement ingress filtering techniques to provide a finer granularity on the control of the source addresses used.

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 April 6, 2012.

Table of Contents

1. Requirements notation

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 [RFC2119].

2. Introduction

This memo describes SEND SAVI (SEcure Neighbor Discovery Source-Address Validation Implementation), a mechanism to provide source address validation for IPv6 networks using the SEND protocol [RFC3971]. The proposed mechanism is intended to complement ingress filtering techniques to provide a finer granularity on the control of the source addresses used.

SEND SAVI uses DAD_NSOL (Duplicate Address Detection Neighbor SOLicitation), DAD_NADV (DAD Neighbor ADVertisement), NUD_NSOL (Neighbor Unreachability Detection Neigbor SOLicitation) and NUD_NADV (NUD Neighbor ADVertisement) messages to validate the address ownership claim of a node. In additon, SEND SAVI uses RADV (Router ADVertisement) messages to identify routers, and therefore restrict the nodes which can generate packets containing off-link IPv6 source addresses. Using the information contained in these messages, host and router IPv6 addresses are associated to layer-2 binding anchors, so that data packets will be validated by checking for consistency in this binding, as described in [I-D.ietf-savi-framework].

Scalability of a distributed SAVI system comprised of multiple SEND SAVI devices is preserved by means of a deployment scenario in which SEND SAVI devices form a "protection perimeter". In this deployment scenario, validation is only performed when the packet ingress to the protection perimeter.

The SEND SAVI specification, as defined in this document, is limited to links and prefixes in which every IPv6 host and every IPv6 router

uses the SEND protocol [RFC3971] to protect the exchange of Neighbor Discovery information.

SEND SAVI is designed to be deployed in existing SEND networks with a minimum set of changes. In particular, SEND SAVI does not require any changes in the nodes whose source address is to be verified. This is due to the fact that verification solely relies in the usage of already available protocols. Therefore, SEND SAVI does neither define a new protocol, nor define any new message on existing protocols, nor require that a host or router uses an existent protocol message in a different way.

An overview of the general framework about Source Address Validation Implementation is presented in [I-D.ietf-savi-framework].

3. Non-normative Background to SEND SAVI

3.1. Address Validation Scope

The application scenario of SEND SAVI is limited to the local link. This means that the goal of SEND SAVI is to verify that the source addresses of the packets generated by the nodes attached to the local link have not been spoofed, and that only legitimate routers generate packets with off-link IPv6 source addresses.

In a link there usually are hosts and routers attached. Hosts generate packets with their own addresses as the source address. This is the so-called local traffic, while routers send packets containing a source address other than their own, since they are forwarding packets generated by other hosts (usually located in a different link). This is the so-called transit traffic.

SEND SAVI allows the validation of the source address of the local traffic, i.e. it allows to verify that the source addresses of the packets generated by the nodes attached to the local link have not been spoofed. In addition, since SEND does provide the means to verify that a node claiming to act as a router is indeed authorized to do so, SEND SAVI also provides means to prevent hosts from generating packets with source addresses derived from off-link prefixes. Note, however, that SEND SAVI does not provide the means

to verify if a given router is actually authorized to forward packets containing a particular off-link source address. Other techniques, like ingress filtering [RFC2827], are recommended to validate transit traffic.

3.2. Binding Creation for SEND SAVI

Filtering is performed according to the binding existing between a layer-2 anchor (the binding anchor) and an IPv6 address. These bindings should allow legitimate nodes to use the bounded IPv6 address as source address, and prevent illegitimate nodes to do so.

SEND [RFC3971] provides tools to assure that a ND (Neighbor Discovery) message containing a CGA option and signed by a RSA option has been generated by the legitimate owner of the CGA IPv6 address. It also provides tools to verify that a Router Advertisement (RADV) message signed by a RSA option with a key bounded to a CGA [RFC3972] or a certificate, has been generated by a legitimate router.

SEND SAVI uses SEND validated messages to create bindings between the binding anchor and the CGA. The events that trigger the binding creation process in a SEND SAVI device are:

- The reception of a DAD_NSOL message, indicating the attempt of a node to configure an address. This may occur when a node configures an address for the first time or after being idle for some time, or when the node has changed the physical attachment point to the layer-2 infrastructure.
- The reception of any other packet (including data packets) with a source address for which no binding exists. This would occur if a DAD_NSOL message was lost, or if a node has changed the physical attachment point to the layer-2 infrastructure without issuing a DAD_NSOL message, a SAVI device loses a binding (for example, due to a restart) or the link topology changes and the SAVI instances through which the packets ingress to the protected perimeter do not have a binding for the node.

When the binding creation process is triggered, the SEND SAVI device has to assure that the node for which the binding is to be created is the legitimate owner of the address. For a binding creation process initiated by a DAD_NSOL exchange, the messages to consider for address ownership validation are validated DAD_NSOL messages arriving

from other locations or a validated DAD_NADV message indicating that other node had configured the address before. For the case in which other packets than a DAD_NSOL initiate the creation of the binding, the SEND SAVI device explicitly requires the node to prove address ownership by issuing a secured NUD_NSOL which has to be answered by a secured NUD_NADV by the probed node.

Bindings are refreshed periodically by means of a secured NUD_NSOL message issued by the SEND SAVI device which has to be answered by a valid NUD_NADV message by the node for which the binding exist.

Validated RADV messages are used to associate router authorization to existing bindings (i.e. to an IPv6 address which is also associated to a binding anchor). In this case, packets with off-link source addresses are only forwarded if they are received with a binding anchor which is associated to the IPv6 address of a router.

SEND SAVI could be sensible to replay attacks, i.e. situations in which a secured SEND message is replayed by a non-legitimate node. For example, if ports are used as binding anchors, a node could immediately re-inject a valid SEND message being received from a legitimate node to force, in the SAVI device to which it is attached to, the creation of a binding for which it is not authorized. While SEND provides some means to prevent the replaying of ND messages, this built-in protection is not enough for SEND SAVI. SEND anti-replay protection relies on the use of nonces to validate

advertisements that were previously solicited, and the use of timestamps to validate solicitation messages and unsolicited advertisements. The emphasis for SEND anti-replay protection is to assure that confidence in some information (for example, the relationship between an IPv6 address and a layer-2 address) is not kept for more time than reasonable. However, in SEND SAVI, information which may be expected to be true for some period, like the relationship of an IPv6 address and a layer-2 address, can be abused to create an illegitimate SAVI binding in a time span shorter than the time reasonable to consider the information aged. As a consequence, SEND SAVI is designed to rely only on messages with a low chance of being replayed:

- Unsolicited DAD_NSOL messages. According to the SEND SAVI specification (Section 4.3.2), these messages can only be forwarded to ports through which a previous binding for the same

IPv6 address existed.

- Valid NUD_NADV messages in response to a secured NUD_NSOL sent by the SEND SAVI device, both exchanged through the same port.

Section 5.2 discusses the resulting protection provided by SEND SAVI against replay attacks.

3.3. SAVI Logging

While the primary goal of SEND SAVI is simply to prevent improper use of IP addresses, a secondary goal is to assist in traceability for determining who an improper actor is. For example, if a remote site reports that a DoS (or component of a DDoS) is coming from the SEND SAVI site, SEND SAVI enforcement can be a useful component in a response.

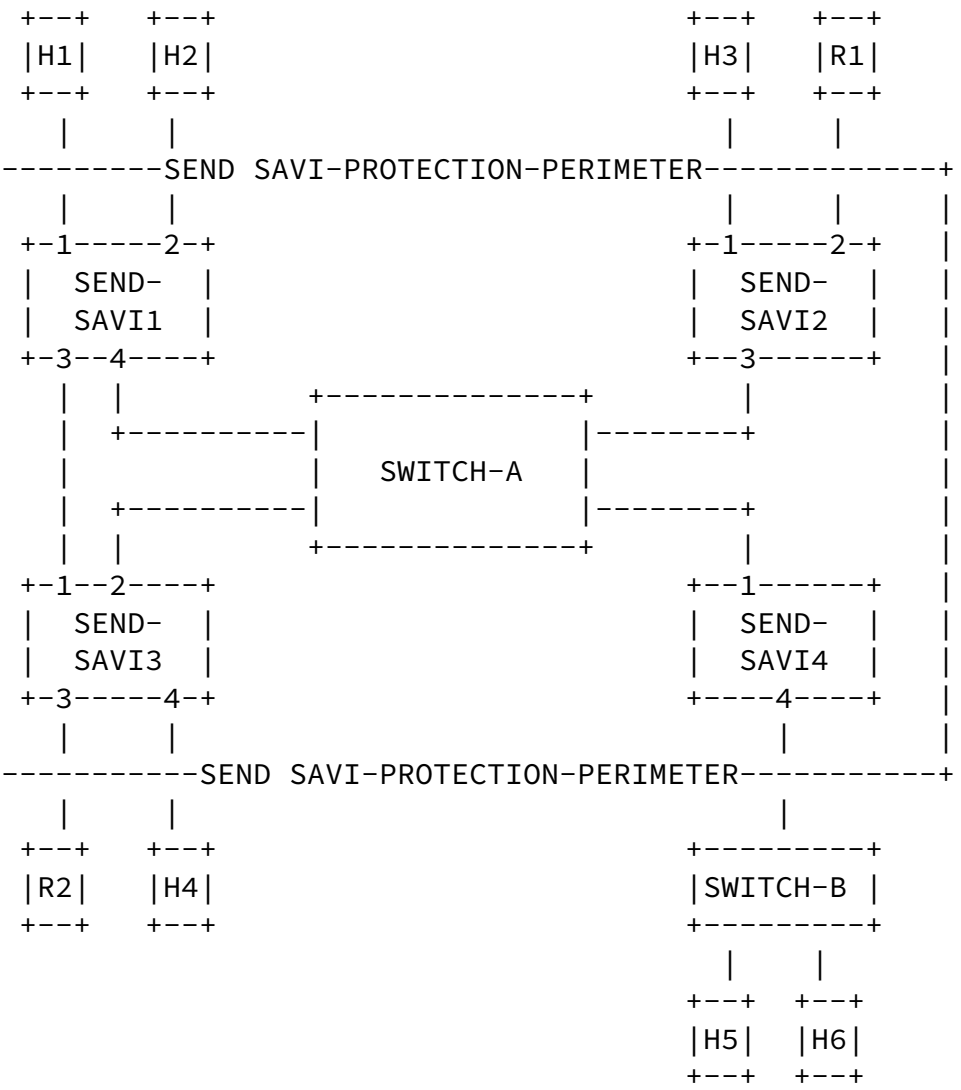
In order to support these and other similar activities, it is a good idea if SAVI devices perform logging of the creation, modification, or removal of address bindings. Any protocol support, such as SYSLOG support for sending those logs to a common server, would be a topic for a future separate document.

3.4. SEND SAVI Protection Perimeter

In order to reduce computing and state requirements in SEND SAVI devices, SEND SAVI devices can form a "protection perimeter" [I-D.ietf-savi-framework]. In this model, source address validation is performed only when packets enter in a protected realm defined through the protection perimeter. The perimeter is defined by appropriate configuration of the roles of each port, which can be 'Validating ports' and 'Trusted ports':

- Validating ports (VPs) are those in which SEND SAVI filtering and binding creation is performed.
- Trusted ports (TPs) are those in which neither SEND SAVI filtering nor binding creation are performed. So, packets received through Trusted ports are not filtered by SEND SAVI. The only SEND messages received through a Trusted port which are processed are those related with certificates, prefix information and Neighbor Advertisements for Duplicate Address Detection (DAD_NADV).

The following figure shows a typical topology involving trusted and untrusted infrastructure.



Trusted ports are used for connections with trusted infrastructure, including the communication between SEND SAVI devices or other trusted nodes.

Port 3 of SEND-SAVI1 and port 1 of SEND-SAVI3 are trusted because

they connect two SAVI devices. Port 4 of SEND-SAVI1, port 3 of SEND-SAVI2, port 2 of SEND-SAVI3 and port 1 of SEND-SAVI4 are trusted because they connect to SWITCH-A to which only trusted nodes are connected.

Validating ports are used for connection with non-trusted infrastructure and with routers. Therefore, hosts are normally connected to Validating ports. Routers are also recommended to be connected to Validating ports. Non-SEND SAVI switches that are outside of the SAVI protection perimeter are also connected through Validating ports. In particular, non-SEND-SAVI devices which connect directly to hosts or which do not have a SEND SAVI capable device between themselves and the hosts, are connected through a Validating port. So, in the figure above, ports 1 and 2 of SEND-SAVI1, port 1 of SEND-SAVI2, port 4 of SEND-SAVI3 are Validating ports because they connect to hosts. Port 2 of SEND-SAVI2 and port 3 of SEND-SAVI3 are Validating ports because they connect to routers. Port 4 of SEND-SAVI4 is also a Validating port because it is connected to SWITCH-B which is a non-SEND-SAVI capable switch which is connected to hosts H5 and H6.

4. SEND SAVI Specification

4.1. SEND SAVI Data Structures

The following three data structures are defined for SEND SAVI operation:

SEND SAVI Data Base. The SEND SAVI function relies on state information binding the source IPv6 address used in data packets to the port through which the legitimate node connects. Such information is stored in the SEND SAVI Data Base. The SEND SAVI Data Base contains one entry for each of the IPv6 source addresses in use on a Validating port of the SEND SAVI device. The SEND SAVI Data Base is populated with the contents of validated SEND messages. Each entry contains the following information:

- IPv6 source address
- Binding anchor, such as Layer-2 address, port through which the packet was received, etc.
- Validating Port through which the packet was received. Note that if the binding anchor used is also the port, the information stored in both elements is the same.
- Lifetime
- Status: TENTATIVE_DAD, TENTATIVE_NUD, VALID, TESTING_VP, TESTING_VP'

- o Alternative binding anchor, to be used when the entry is in TESTING_VP' state
- o Alternative Validating port, VP', to be used when the entry is in TESTING_VP' state
- o Creation time: the value of the local clock when the entry was firstly created

SEND SAVI Prefix list. SEND SAVI devices need to know which are the link prefixes, in order to identify local and off-link traffic. A SEND SAVI device MUST be able to obtain this information from validated RADV messages, either coming from Validating or Trusted ports, as described in Section 4.3.2. This information is not specific to a given port. The SEND SAVI Prefix list contains one entry per prefix in use, as follows:

- o Prefix
- o Lifetime

When the SEND SAVI device boots, it MUST send a secured RSOL message. The SAVI device SHOULD issue a secured RSOL in case the prefix entry is about to expire.

SEND SAVI Router list. SEND SAVI keeps a table with one entry for each authorized router in use connected to a Validating port of the SAVI device. This entry is created for the IPv6 source address from which a validated RADV message addressed to the all-nodes multicast address or to the IPv6 address of the SEND SAVI device has been received from a Validating port. The SAVI device SHOULD issue a secured RSOL through the Validating port through which the router is reachable (according to the information stored in the SEND SAVI Data Base) in case the entry is about to expire, in order to ensure that the node is still a router. The information stored in the table is the following:

- o Router IPv6 address. There MUST be an entry in the SEND SAVI Data Base for the same IPv6 address. If the corresponding entry in the SEND SAVI Data Base expires, the entry in this table MUST be removed.
- o Lifetime

4.2. SEND SAVI Device Configuration

In order to perform SEND SAVI operation, some basic parameters of the SEND SAVI device have to be configured. Since a SEND SAVI device operates as a SEND node to generate NUD_NSOL, RSOL or CPS message, it

- o MUST be configured with a valid CGA address. Note that when the SEND SAVI device configures this address, it MUST behave as regular SEND node, i.e. using secured NSOL messages to perform

DAD, etc.

- MUST be configured with at least one Trust anchor to validate the Certification Paths that is used to validate router information.
- MAY be configured with Certification Paths. The alternative is obtaining them by means of issuing Certification Path Solicitation messages, as detailed in the SEND specification [RFC3971].

In addition, the port role for each port of the SEND SAVI device SHOULD be configured. The guidelines for this configuration are specified in Section 4.4. Unconfigured ports MUST be labeled as Validating ports; in this case performance may be degraded, as discussed in [I-D.ietf-savi-framework].

4.3. Traffic Processing

In this section we describe how packets are processed by a SEND SAVI device. Behavior varies depending on if the packet belogn to local or transit traffic. This is determined by checking if the prefix of the source address is included in the SEND SAVI Prefix List (local traffic) or not included (transit traffic).

4.3.1. Transit Traffic Processing

Transit traffic processing occurs as follows:

- If the transit traffic packet is received through a Trusted port, the data packet is forwarded and no SAVI processing performed.
- If the transit traffic packet is received through a Validating port, the packet is only forwarded if the binding anchor for the packet is associated to the binding anchor of an IPv6 address for which an entry in the Router list exists. If transit traffic is received from a Validating port with a binding anchor which is not associated to an entry in the SEND SAVI Router list, the SEND SAVI device SHOULD discard the packets, and MAY send a RSOL message to the all-routers multicast address to the port though which the packet was received.

4.3.2. Local Traffic Processing

If the verification of the source address of a packet shows that it belongs to local traffic, this packet is processed using the state

machine described in this section. SEND SAVI is designed to perform source address validation for both hosts and routers, so in the following description we will refer in general to nodes.

For the rest of the section, the following assumptions hold:

- When it is stated that a secured NUD_NSOL message is issued by a SEND SAVI device through a port P, this means the following: the SEND SAVI device generates NUD_NSOL messages according to the Neighbor Unreachability Detection procedure described in

[RFC4861], addressed to the IPv6 target address (source address of the packet triggering the procedure). These messages are secured by SEND as defined in [RFC3971]. The source address used for issuing the NUD_NSOL is the source address of the SEND SAVI device. The message is sent only through port P.

- When it is stated that a validated NUD_NADV message is received by a SEND SAVI device, this means that: a SEND secured NUD_NADV message has been received by the same port P through which the corresponding NUD_NSOL message was issued, and the NUD_NADV message has been validated according to [RFC3971] to prove ownership for the IPv6 address under consideration and to prove that it is a response for the previous NUD_NSOL message issued by the SEND SAVI device (containing the same nonce value as the NUD_NSOL message to which it answers).

We use VP to refer to a Validating port, and TP to refer to a Trusted port.

The state machine is defined for a binding of a given source IP address in a given SEND SAVI device. In the transitions considered, packets described as inputs refer to the IPaddr IPv6 address associated to the state machine.

The possible states for a given IPaddr are: NO_BIND, TENTATIVE_DAD, TENTATIVE_NUD, VALID, TESTING_VP and TESTING_VP'. The NO_BIND state represents that no binding exists for IPaddr; this is the state for all addresses unless a binding is explicitly created.

The states can be classified into forwarding states, i.e. states in which packets with the binding anchor associated to the IPv6 address are forwarded, and non-forwarding states, i.e. states in which packets coming from the binding anchor associated to the IPv6 address

different to the ones used for signaling are not forwarded. VALID, TENTATIVE_DAD, TESTING_VP and TESTING_VP' are forwarding states, while NO_BIND and TENTATIVE_NUD are non-forwarding states.

The state machine defined for SEND SAVI operation adheres to the following design guidelines:

- The only events which trigger state changes from forwarding to non-forwarding states and vice versa are the reception of DAD_NSOL, DAD_NADV and NUD_NADV, or the expiration of a timer.The other possible input to consider is 'any other packet', which could generate changes to states belonging to the same forwarding or non-forwarding class as the original state (i.e. when 'any other packet' is received, the state cannot move from being forwarding to non-forwarding and vice versa). A special case of 'any other packet' is when validated RADV are received, which can result in the update of the SEND SAVI Prefix or Router lists.

  Note that non-validated SEND messages always belong to the 'any other packet' category. The reduced set of messages being able to trigger a change simplifies the processing at SEND SAVI devices.
- DAD_NADV and NUD_NADV are only processed when they are a response to a DAD_NSOL or a NUD_NSOL message.
- ND messages are only used by SEND SAVI devices if they are valid. If any of the ND messages used by SEND SAVI is not valid, it is discarded. SEND SAVI devices SHOULD assume that such messages received from Trusted ports have been validated by other SEND SAVI devices, so they SHOULD NOT validate them in order to reduce processing load at the SEND SAVI device.
- The only messages the SEND SAVI device is required to generate for SEND SAVI operation are NUD_NSOL messages. This also simplifies the state machine.
- Well-behaved nodes are expected to initiate communication by sending secured DAD_NSOL messages. The SEND SAVI state machine is tailored to efficiently process these events. The reception of other packet types without receiving previously validated DAD_NSOL messages is assumed to be consequence of bad-behaving nodes or infrequent events (such as packet loss, a change in the topology connecting the switches, etc.) While a binding will ultimately be created for nodes affected by such events, simplicity of the state machine is prioritized over any possible optimization for these cases.
- If a node has an address configured, and it can prove the

ownership of this address, the binding is preserved regardless of any indication that a binding for the same source address could be configured in other SEND SAVI device. Bindings for the same source address in two or more SEND SAVI devices may occur due to several reasons, for example when a host moves (the two bindings exist just for a short period of time), or when many nodes generate the same address and the DAD procedure has failed. In these infrequent cases, SEND SAVI preserves connectivity for the resulting bindings.

The SEND SAVI device MUST join the Solicited Node Multicast group for all the addresses which state is other than NO_BIND. This is needed to make sure that the SEND SAVI device receives DAD_NSOL messages issued for those addresses. Note that it may not be enough to relay on the IGMP messages being sent by the node behind the Validating port, for which a binding for the corresponding address exist, since the node may move and after a while, and the packets for that particular Solicited Node Multicast group will no longer be forwarded to the SEND SAVI device.

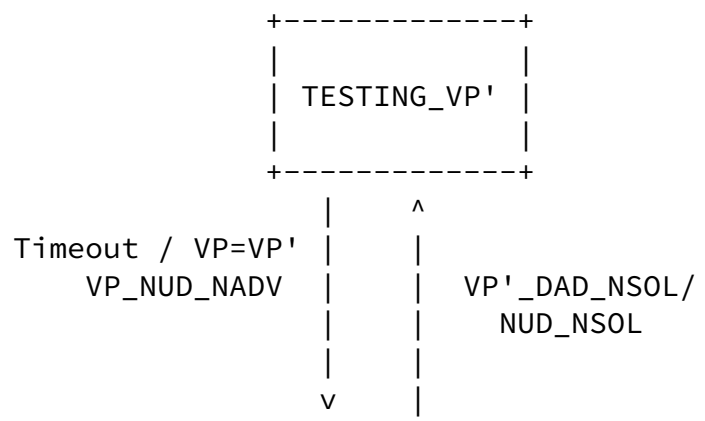
SEND SAVI devices MUST be able to use validated CPA messages, sent in reply to CPS messages, to acquire certificates used to validate ND messages. In order to process a CPA message received from a

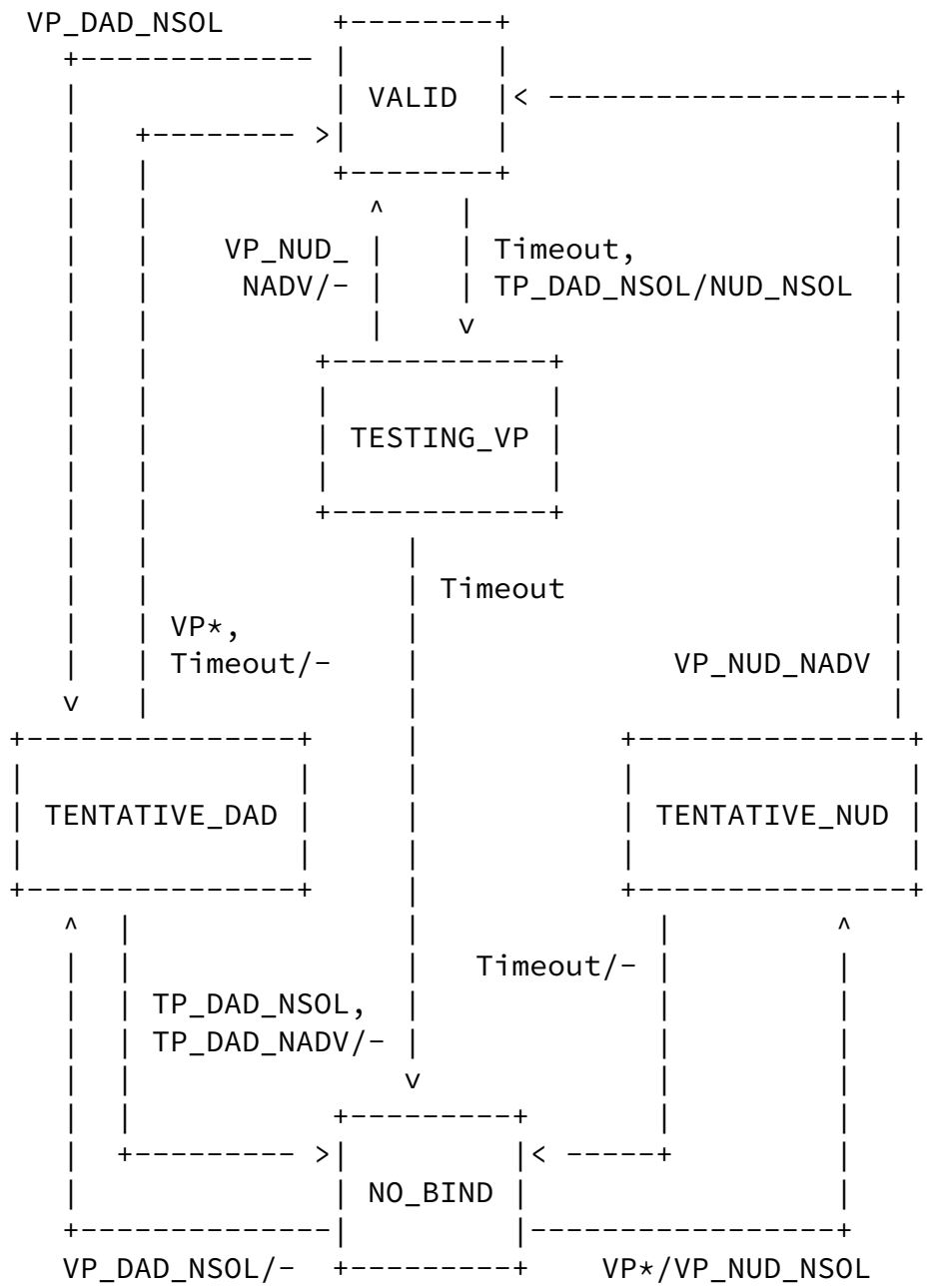
Validating port, an entry for the source address of the message MUST exist in the SEND SAVI Data Base. CPA messages received from Trusted ports are always checked and processed.

SEND SAVI devices MUST use validated RADV messages to update the SEND SAVI Prefix list and the SEND SAVI Router list. SEND SAVI devices MAY only consider for this purpose (updating SEND SAVI Prefix and Router lists) RADV messages addressed to either its own IPv6 address or to the all-nodes multicast address. Validated RADV messages received from Trusted ports MUST be used to update accordingly the SEND SAVI Prefix and Router lists in the SEND SAVI device. Validated RADV messages received from Validating ports MUST be processed according to the specific rules defined in the state machine for local traffic processing. In short, RADV messages received from Validating ports are only processed for updating the SEND SAVI Router and Prefix lists if a binding for the source IPv6 address of the RADV message is in a forwarding state.

We next describe how different inputs are processed depending on the state of the binding of the IP address 'IPaddr'. A Waiting_lifetime timer is associated to each binding.

A simplified version is depicted in the next figure (note that all ND messages are assumed to be validated):





Simplified SEND SAVI state machine

NO_BIND

When the node is in this state, there are no unresolved DAD_NSOL or NUD_NSOL messages (generated by SEND SAVI), so the only relevant

inputs are DAD_NSOL messages coming either from VP or TP, or any packet other than DAD_NSOL coming from VP or TP. There are no timers configured for this state.

- If a DAD_NSOL message is received from a Validating port VP, the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. If the message is valid, then the SEND SAVI device forwards this message to all appropriate Trusted ports (the subset of Trusted ports which belong to the forwarding layer-2 topology, with the restrictions imposed by the MLD snooping mechanism, if applied). DAD_NSOL messages are not sent through any of the ports configured as Validating Ports. The SEND SAVI device sets the Waiting_timer to TENT_LT, stores all the information required for future validation of the corresponding DAD_NADV message (such as the nonce of the message), creates a new entry in the SEND SAVI Data Base for IPaddr and the binding anchor of the received DAD_NSOL, and changes the state to TENTATIVE_DAD. Creation time is set to the current value of the local clock. Note that in this case it is not possible to check address ownership by sending a NUD_NSOL because while the node is waiting for a possible DAD_NADV its address is in tentative state and the node cannot respond to NSOL messages [RFC4862].
- If any packet other than a DAD_NSOL is received through a Validating port VP, the SEND SAVI device issues a secured NUD_NSOL through port VP. The SEND SAVI device sets the Waiting_timer to TENT_LT. The SEND SAVI device creates a new entry in the SEND SAVI Data Base for IPaddr and the binding anchor of the received packet, and the state is changed to TENTATIVE_NUD. Creation time is set to the current value of the local clock. The SAVI device MAY discard the packet while the DAD procedure is being executed, or MAY store it in order to send it if the next transitions are (strictly) TENTATIVE_NUD and then VALID.
- If a DAD_NSOL message containing IPaddr as the target address is received through a Trusted port, the SEND SAVI device SHOULD assume that the message has been validated. This message is not forwarded through any of the Validating ports but they are sent through the proper Trusted Ports (as defined by the switch behavior that will depend on whether it performs MLD snooping or not). The state is not changed.
- Any packet other than a DAD_NSOL received from a Trusted port is forwarded to its destination. This packet is assumed to come from a SEND SAVI device that has securely validated the binding according to SEND SAVI rules (unless the SEND SAVI perimeter has been breached). The state is not changed.

TENTATIVE_DAD

To arrive to this state, the SEND SAVI device has received a validated DAD_NSOL coming from port VP and it has forwarded it to the appropriate TPs. The relevatn events occurring in this state are: the reception of a DAD_NADV message from a TP, a DAD_NSOL message from VP, other Validating port VP' or TP, a data packet from VP, and the expiration of the timer initiated when the DAD_NSOL was received at port VP.

- If a DAD_NADV is received from a Trusted port, the SEND SAVI device SHOULD assume that the message has been validated. The reception of a valid DAD_NADV message indicates that the binding cannot be configured for port VP. The state is changed to NO_BIND, and the Waiting_timer cleared.
- If a DAD_NSOL is received from a Trusted port, the SEND SAVI device SHOULD assume that the message has been validated. The reception of a valid DAD_NSOL indicates that a node connected to another SEND SAVI device may be trying to configure the same address at the same time. The DAD_NSOL message is forwarded to port VP, so that the node at port VP will not configure the address, as stated in [RFC4862]. The DAD_NSOL message is also forwarded to all appropriate Trusted ports. Then, the Waiting_timer is cleared, and the state is changed to NO_BIND.
- Any packet other than a validated DAD_NSOL or DAD_NADV received from a Trusted port is forwarded to its destination. This packet is assumed to come from a SEND SAVI device that has securely validated the binding according to SEND SAVI rules (unless the SEND SAVI perimeter has been breached). The state is not changed.
- If a DAD_NSOL is received from a Validating port VP' different to VP, the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. The reception of a valid DAD_NSOL from port VP' indicates that a node connected to VP' may be trying to configure the same address at the same time. The DAD_NSOL message is forwarded to port VP, so that the node at port VP will not configure the address, as stated in [RFC4862]. The DAD_NSOL message is also forwarded to all appropriate Trusted ports. Then, the entry in the SEND SAVI Data Base for IPaddr is updated with the binding anchor of the DAD_NSOL received from port VP', the Waiting_timer is set to TENT_LT, and the state remains in TENTATIVE_DAD, although in this case with VP=VP'.
- Any other packet than a validated DAD_NSOL is received from a Validating port VP' different from VP is discarded. The state is not changed.
- If a DAD_NSOL is received from port VP, the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. If the DAD_NSOL message is valid, the Waiting_timer is set to TENT_LT, and the state remains in TENTATIVE_DAD.

- o If any packet other than a DAD_NSOL is received from VP, it is assumed that the node has configured its address, although it has done it in less time than expected by the SEND SAVI device (less than TENT_LT). Since the node proved address ownership by means of the validated DAD_NSOL message, the Waiting_timer is set to DEFAULT_LT, and the state is changed to VALID.
  A particular case occur if the packet received is a RADV message. The RADV message is checked for validity, and it is discarded if it is not valid (and the Waiting_timer is not changed, and the state reamins in TENTATIVE_DAD). If it is valid, the message is forwarded to the appropriate Trusted ports. In addition, either an entry for this IPv6 source address in the SEND SAVI Router List is created, or the lifetime of an existing entry is updated with the information received in this message. The SEND SAVI Prefix list MUST also be updated according to the content of the RADV message. The SEND SAVI device MAY not process (although it MUST forward) RADV messages addressed to destinations other than the all-nodes multicast address or to the IPv6 address of the SEND SAVI device.
- o If Waiting_timer expires, it is assumed that no other node has configured this address. Therefore, the Validating port VP could be bound to this IPv6 address. The Waiting_timer is set to DEFAULT_LT, and the state is changed to VALID.

VALID

To arrive to this state, successful validation of address ownership has been completed and a binding for IPaddr has been created. Relevant transitions for this state are triggered by the reception of DAD_NSOL from ports VP, VP' or TP, and any packet other than DAD_NSOL from VP' or TP. The expiration of Waiting_timer is also relevant to trigger a check for address ownership for the node at VP.

- o If a DAD_NSOL with IPaddr as source address is received through Validating port VP, the message is checked for validity. If the message is not valid, it MUST be discarded. If the message is valid, it is forwarded to the appropriate Trusted ports. The Waiting_timer is set to TENT_LT and the state is changed to TENTATIVE_DAD.
- o Any packet other than a DAD_NSOL containing IPaddr as a source address arriving from Validating port VP is forwarded

appropriately. The state is not changed.
A particular case occur if the packet received is a RADV message. The RADV message is checked for validity, and it is discarded if it is not valid. If it is valid, the message is forwarded to the appropriate Trusted ports. In addition, either an entry for this IPv6 source address in the SEND SAVI Router List is created, or the lifetime of an existing entry is updated with the information received in this message. The SEND SAVI Prefix list MUST also be

updated according to the content of the RADV message. The SEND SAVI device MAY not process (although it MUST forward) RADV messages addressed to destinations other than the all-nodes multicast address or to the IPv6 address of the SEND SAVI device.

- If a DAD_NSOL with IPaddr as source address is received through a Trusted port, the SEND SAVI device SHOULD assume that the message has been validated. The message is forwarded to VP. The Waiting_timer is set to TENT_LT, a secured NUD_NSOL message is sent to IPaddr through VP and the state is changed to TESTING_VP.
- If any packet other than a DAD_NSOL with IPaddr as source address is received through a Trusted port, the packet is forwarded to VP and to other appropriate Trusted ports. A secured NUD_NSOL is sent to VP, the Waiting_timer is set to TENT_LT, and the state is changed to TESTING_VP.
- If a DAD_NSOL packet with IPaddr as source address is received through a Validating Port VP' (VP' different from the current Validating port for this binding), the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. If the message is valid, the message is forwarded to VP. In addition, a secured NUD_NSOL is sent to VP, the binding of the DAD_NSOL received from VP' is stored in the Alternative Binding Anchor for future use if the node at VP' is finally selected, the Alternative Validating Port is set to VP', the Waiting_timer is set to TENT_LT, and the state is changed to TESTING_VP'.
- If any packet other than a DAD_NSOL with IPaddr as source address is received from a Validating port VP', different from the current Validating port for this binding, VP, the packet is discarded. The SEND SAVI device MAY issue a secured NUD_NSOL through port VP, store the binding of the DAD_NSOL received from VP' in the Alternative Binding Anchor for possible future use, set the Alternative Validating Port to VP', set the Waiting_timer to TENT_LT, and change the state to TESTING_VP'. An alternative to this behavior is that the SEND SAVI device MAY not do anything (in

this case, the state would eventually change after a maximum DEFAULT_LT time, if the node at VP does not respond to a NUD_NSOL at TESTING_VP, the state is moved to NO_BIND). Then a packet arriving from VP' would trigger a process that may end up with binding for the node connecting to VP'.

- If Waiting_timer expires, a secured NUD_NSOL message is sent through port VP to IPaddr, the Waiting_timer is set to TENT_LT, and the state is changed to TESTING_VP. In the TESTING_VP state packets are still being forwarded until the timer expires without receiving a NUD_NADV.

TESTING_VP

When the SEND SAVI device enters in the TESTING_VP state, the current Validating port is under check through a secured NUD_NSOL message

generated by the SEND SAVI device. While testing, packets from the current Validating port are forwarded. Packets coming from Trusted ports are also forwarded. The relevant events for this state are the reception of a NUD_NADV message from VP, the reception of a DAD_NSOL message from VP, VP' or TP, the reception of any packet other than the previous cases from VP, VP' or TP, and the expiration of the timer associated to the reception of NUD_NADV.

- If a NUD_NADV packet is received from VP, the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. If the message is valid, the Waiting_timer is changed to DEFAULT_LT, and the state is changed to VALID. The message is not forwarded to any other port.
- If a DAD_NSOL message is received from VP, the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. If the message is valid, it is forwarded to the appropriate Trusted ports, the Waiting_timer is set to DEFAULT_LT, and the state is changed to TENTATIVE_DAD.
- If a RADV packet is received from VP, the message is checked for validity, and it is discarded if it is not valid. If it is valid, the message is forwarded appropriately. Either an entry for this IPv6 source address in the SEND SAVI Router List is created, or the lifetime of an existing entry is updated with the information received in this message. The SEND SAVI Prefix list MUST also be updated according to the content of the RADV message. The SEND SAVI device MAY ignore and discard RADV messages addressed to destinations other than the all-nodes multicast address or to the

IPv6 address of the SEND SAVI device. The state remains in TESTING_VP. Note that if the timeout expires later, while still in the TESTING_VP state, the entry of the SEND SAVI Router List will also be removed.

- Any packet other than DAD_NSOL or NUD_NADV containing IPaddr as a source address arriving from Validating port VP is forwarded. Neither the Waiting_timer nor the state are changed.
- If a DAD_NSOL packet is received from a Trusted port, the SEND SAVI device SHOULD assume that the message has been validated. The message is forwarded to VP and the appropriate Trusted ports. Neither the Waiting_timer nor the state are changed. The node at VP port is under check: if it still is at port VP, it should answer with a NUD_NADV, and also with a DAD_NADV. If it is not there, neither the NUD_NADV nor the DAD_NADV will be received, the timer will expire, the local state will move to NO_BIND, and the state at the remote node will change to VALID.
- If a packet other than a DAD_NSOL arrives from a Trusted port, the packet is forwarded. Neither the Waiting_timer nor the state are changed.
- If a DAD_NSOL is received from a Validating port VP', the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. If it is valid, the message is forwarded to

VP and to the appropriate Trusted ports. In addition, a secured NUD_NSOL is sent to VP, the binding of the DAD_NSOL received from VP' is stored in the Alternative Binding Anchor for future use if the node at VP' is finally selected, the Alternative Validating Port is set to VP', the Waiting_timer is set to TENT_LT, and the state is changed to TESTING_VP'.

- Any other packet received from a Validating port VP' is discarded. This may occur because the node has moved but have not issued a DAD_NSOL or the DAD_NSOL message has been lost. The state will eventually move to NO_BIND, and then the packets sent from VP' will trigger the creation of the binding for VP'.
- If the Waiting_timer expires, the Waiting_timer is cleared and the state is changed to NO_BIND.

TESTING_VP'

To arrive to this state an indication that a node at VP' wants to send data with IPaddr as source address occurred while a binding existed for VP. The binding anchor of the packet received through

VP' which triggered the change of the state to TESTING_VP' was stored, so that it can be retrieved if the node at VP' is determined as the legitimate owner of IPaddr. The SEND SAVI device has issued a NUD_NSOL to IPaddr through port VP. The relevant events that may occur in this case are the reception of a NUD_NADV from port VP, the reception of DAD_NSOL from VP, VP', TP and VP" (VP" different from VP and VP'), the reception of any other packet from VP, VP', TP or VP", and the expiration of the timer.

- If a NUD_NADV is received from port VP, the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. The reception of a valid NUD_NADV indicates that the node at VP is defending its address. The binding anchor in use is kept, VP is kept as the Validating port, the Waiting_timer is set to DEFAULT_LT, and the state is changed to VALID.
- If a DAD_NSOL is received from port VP, the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. If the message is valid, it is forwarded to VP'. The binding anchor in use is kept, VP is kept as the Validating port, the Waiting_timer is set to TENT_LT and the state is changed to TENTATIVE_DAD. When the DAD_NSOL message is received by the node at VP', this node is expected to unconfigure its address.
- If a RADV message is received from VP, it is checked for validity, and it is discarded if it is not valid. If it is valid, the message is forwarded appropriately. Either an entry for this IPv6 source address in the SEND SAVI Router List is created, or the lifetime of an existing entry is updated with the information received in this message. The SEND SAVI Prefix list MUST also be updated according to the content of the RADV message. The SEND SAVI device MAY ignore and discard RADV messages addressed to

destinations other than the all-nodes multicast address or to the IPv6 address of the SEND SAVI device. The state remains in TESTING_VP' and the Waiting_timer is left unchanged. Note that if the timeout expires later, while still in the TESTING_VP' state, the entry of the SEND SAVI Router List will also be removed.

- Any packet other than a validated DAD_NSOL, a validated NUD_NADV or a validated RADV coming from port VP, is forwarded, and the state is not changed.
- If a DAD_NSOL is received from port VP', the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. If the message is valid, it is forwarded to VP. The binding anchor in use is kept, the alternative binding anchor is

set to the binding anchor of the DAD_NSOL received from port VP', VP' is kept as Alternative Validating Port, VP is kept as the Validating port, the Waiting_timer is set to DEFAULT_LT, and the state is not changed.

- Any packet other than a DAD_NSOL coming from port VP is discarded, and the state is not changed.
- If a DAD_NSOL is received from port VP", different from VP and VP', the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. If the message is valid, it is forwarded to VP and VP'. VP' is expected to unconfigure its address if the message triggering the transition to this state was a VP'_DAD_NSOL message (and not any other packet). The state remains in TESTING_VP' although the binding of the DAD_NSOL received from VP" is stored in the Alternative Binding Anchor for future use if the node at VP" is finally selected, and the Alternative Validating Port is set to VP". The Waiting_timer is not changed.
- Any packet other than a DAD_NSOL received from port VP" is discarded and does not affect to the state.
- If a DAD_NSOL is received from a Trusted port, the SEND SAVI device SHOULD assume that the message has been validated. Then, the message is forwarded to ports VP, VP' and other appropriate Trusted ports. The Waiting_timer is left unchanged and the state is changed to TESTING_VP. VP' is expected to unconfigure its address if the packet triggering the transition to this state was a VP'_DAD_NSOL message.
- Any packet other than a DAD_NSOL coming from a Trusted port is forwarded appropriately, but the state is not changed.
- If Waiting_timer expires, it is assumed that the node for which the binding existed is no longer connected through port VP. Therefore, the Validating port VP' could be bound to this IPv6 address. The Waiting_timer is set to DEFAULT_LT and the state is changed to VALID.

TENTATIVE_NUD

To arrive to this state, a data packet has been received through port VP without any existing binding in the SEND SAVI device. The SEND SAVI device has sent a NUD_NSOL message to VP. The relevant events for this case are the reception of a NUD_NADV from port VP, the reception of DAD_NSOL from port VP, VP' or TP, and the reception of

any packet other than DAD_NSOL and NUD_NADV for port VP, and different from DAD_NSOL for VP' or TP. In addition, the Waiting_timer may expire.

- If a NUD_NADV message is received through port VP, the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. If the message is valid, the Waiting_timer is set to TENT_LT, and the state is changed to VALID. The message is not forwarded to any port.
- If a DAD_NSOL message is received through port VP, the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. If the message is valid, it is forwarded to the appropriate Trusted ports, the Waiting_timer is set to TENT_LT and the state is changed to TENTATIVE_DAD.
- Any packet other than NUD_NADV or DAD_NSOL received through port VP is discarded.
- If a DAD_NSOL message is received through port VP' different from port VP, the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. If the message is valid, it is forwarded to the appropriate Trusted ports, the Waiting_timer is set to TENT_LT, the binding anchor of the DAD_NSOL message received from port VP' is set as the binding anchor, the Validating Port set to VP', and the state is changed to TENTATIVE_DAD.
- Any packet other than DAD_NSOL received through port VP' MUST NOT be forwarded unless the next state for the binding is VALID. The packets received MAY be discarded or MAY be stored for being sent if the state changes later to VALID. The state is left unchanged.
- If a DAD_NSOL message is received through a Trusted port, the SEND SAVI device SHOULD assume that the message has been validated. The message is forwarded to port VP, and the state is left unchanged.
- Any other packet received from a Trusted port is forwarded appropriately. This packet may come from a SEND SAVI device that has securely validated the attachment of the node to its Validating port according to SEND SAVI rules. The state is left unchanged.
- If Waiting_timer expires, the Waiting_timer is cleared and the state is changed to NO_BIND.

4.4. SEND SAVI Port Configuration Guidelines

The detailed guidelines for port configuration in SEND SAVI devices are:

- Ports that are connected to another SEND SAVI devices SHOULD be configured as Trusted ports. Not doing so will increase significantly the CPU time, memory consumption and signaling traffic due to SEND SAVI validation, in both the SEND SAVI devices and the node whose address is being validated.
- Ports connected to hosts SHOULD be configured as Validating ports. Not doing so will allow the host connected to that port to send packets with spoofed source address.
- Ports connected to routers SHOULD be configured as Validating ports. However, the SEND SAVI specification also allows the routers to be connected to Trusted ports, as they are assumed to be part of the trusted infrastructure. When connected through a Trusted port, a router can generate traffic with any source address, even those belonging to the link, while when connected through a Validating port it can only send traffic using off-link source addresses, or its own source addresses. When routers are connected to Validating, authorization for the routing function is bound to the binding anchor of the router itself, instead of being bound to a port configured in a switch.
- Ports connected to a chain of one or more legacy switches that have hosts connected SHOULD be configured as Validating ports. Not doing so will allow the host connected to any of these switches to send packets with spoofed source address.
- Ports connected to a chain of one or more legacy switches that have other SEND SAVI devices but had no routers or hosts attached to them SHOULD be configured as Trusted ports. Not doing so will significantly increase the memory consumption in the SEND SAVI devices and increase the signaling traffic due to SEND SAVI validation.
- Ports connected to a chain of one or more legacy switches that have hosts and possibly a mix of SEND SAVI devices and/or routers, SHOULD be configured as Validating ports. Not doing so will allow the host connected to that port to send packets with spoofed source address.

4.5. VLAN Support

In the case the SAVI device is a switch that supports VLANs, the SAVI implementation will behave as if there was one SAVI process per VLAN. The SAVI process of each VLAN will store the binding information corresponding the nodes attached to that particular VLAN.

4.6. Protocol Constants

TENT_LT is 500 milliseconds.

DEFAULT_LT is 5 minutes.

5. Security Considerations

It should be noted that any SAVI solution is as strong as the binding anchor that it uses. In particular, if the binding anchor is forgeable, then the resulting SAVI solution will be weak. For example, if the binding anchor is a MAC address that can be easily spoofed, then the resulting SAVI will not be stronger than that. On the other hand, if we use switch ports as binding anchors (and there is only one node connected to each port) it is likely that the resulting SAVI solution will be considerably more secure.

SEND SAVI performs its function by binding an IP source address to a binding anchor. If the attacker manages to send packets using the binding anchor associated to a given IP address, SEND SAVI validation will be successful and the SEND SAVI device will allow the packet through. This can be achieved by spoofing the binding anchor or because the binding anchor is shared among the legitimate owner of the address and the attacker. An example of the latter is the case where the binding anchor is a port of a switched network and a legacy switch (i.e. no SEND SAVI capable switch) is connected to that port. All the source addresses of the nodes connected to the legacy switch will share the same binding anchor (i.e. the switch port). This means that nodes connected to the legacy switch can spoof each other's IP address and this will not be detected by the SEND SAVI device. This can be prevented by not sharing binding anchors among nodes.

SEND SAVI is defined to operate only with validated SEND messages. The interaction in a mixed scenario comprising SEND and non-SEND devices should be addressed in other document. However, nodes MUST NOT assume that all SEND messages received from a SEND SAVI device are validated, since these devices only validate the messages strictly required for SEND SAVI operation. Among the number of messages which are not validated, we can name NUD_NSOL messages generated by other nodes and its responses, or RSOL messages.

SEND SAVI improves protection compared to conventional SAVI, as a result of the increased ability of SEND nodes to prove address ownership.

A critical security consideration regarding to SEND SAVI deals with the need of proper configuration of the roles of the ports in a SEND

SAVI deployment scenario. Regarding to security, the main requirement is that ports defining the protected perimeter SHOULD be configured as Validating ports. Not doing so will generate security breaches through which an attacker could send packets using any source address, regardless of the bindings established in other SEND SAVI devices.

5.1. Protection Against Replay Attacks

One possible concern about SEND SAVI is its behavior when an attacker tries to forge the identity of a legitimate node by replaying messages. Note that information that can be valid for SEND a short period after being generated (such as the binding between an IPv6 address and a layer-2 MAC address), is not valid for SEND SAVI if the binding anchor used is the port and the message arrives from the port to which a node replaying a message to use an address illegitimately is connected. For example, with the values recommended by [RFC3971] for TIMESTAMP_FUZZ and TIMESTAMP_DRIFT, a node receiving a DAD_NSOL message, would not discard replays of this message being received within a period of aproximately 2 seconds (more precisely, 2/0.99 seconds). An attacker could replay this message to abort the configuration of an address for a legitimate node, and to gain the right to use the address for LIFETIME_LT seconds. We now discuss the risks of such replay attacks and the protection provided by SEND SAVI.

To perform a security analysis of such a replay attack for SEND SAVI, we have to consider two different cases:

- When the information replayed is tied to the anchor binding, especially if the anchor binding being used is the port through which packets are received. In this case, all the messages which can be create a SEND SAVI binding may be sensible to the replay of valid SEND messages. SEND SAVI creates and maintains bindings as a result of the reception of DAD_NSOL messages and of the exchange of NUD_NSOL/NUD_NADV messages.
- When the information replayed is not tied to the anchor binding (eg. to ports) in SEND SAVI operation. Such situations are the reception of CPA messages containing certificates, or the processing of an RADV message coming from a Trusted port, which can be used in SEND SAVI to populate the SEND SAVI Prefix list. In this case, the security risks are equivalent to those of SEND operation.

A special case is the processing of a RADV message coming from a Validating port. Although part of the information obtained (the router condition of the node connecting to the port) is (indirectly) associated to the anchor binding, in this case the replay of the RADV message does not provide an advantage to an attacker. SEND SAVI requires a binding to exist (between the IPv6 address and the binding anchor) prior to consider the RADV message, so protecting the binding also protects the ability of an attacker to become a router.

We now discuss how replay attacks for DAD_NSOL messages and of the exchange of NUD_NSOL/NUD_NADV messages. For this discussion we assume that the anchor binding in use is the port through which

packets are received.

- To prevent DAD_NSOL replay attacks, DAD_NSOL messages are not forwarded to ports through which an existing binding existed. Therefore, to capture a message that could be used to launch a replay attack, an attacker has to be located either in the port through which the legitimate node is connected (in which case the attack is useless), or in a port in which a legitimate node was before, but it is not now, and for which a binding still exists. For this latter case, an attacker can prevent the configuration of a binding for a legitimate node in other port to which the node could have moved. This risk is inherent to allow layer-2 node mobility in an scenario in which many nodes can attach to the same port (either at the same time or in instants very close one to the other). Another consideration is that this situation reflect the fact that it is impossible to determine the legitimacy of a node without the more secure NUD_NSOL/NUD_NADV exchange, which cannot be used when the nodes claim to be configuring the address.
- When a NUD_NSOL/NUD_NADV exchange is used to create or maintain a state, the messages are only forwarded to the port in which the node claiming to be legitimate is located. In this case, an attacker connected to the same port as the legitimate node can capture either the NUD_NSOL or the NUD_NADV message in order to replay it, but this is useless. The replay of NUD_NSOL is useless, since this message is not used to trigger the creation of a binding unless the SEND SAVI device had previously issued the corresponding NUD_NSOL. The replay of a NUD_NADV message through the same port is useless, since SEND SAVI does not protect against spoofers attached to the same port. Finally, the replay of a

NUD_NADV message through a different port does result neither in the creation of a binding in other SEND SAVI device, nor in the binding created in the SEND SAVI device originating the NUD_NSOL message, since SEND SAVI devices only consider NUD_NADV message received from the same port through which the NUD_NSOL message was sent.

5.2. Protection Against Denial of Service Attacks

The attacks against the SAVI device basically consist on making the SEND SAVI device to consume its resource until it runs out of them, or to slow CPU operation. For instance, a possible attack would be to create state for different addresses in order to waste memory. At some point the SEND SAVI device runs out of memory and it needs to decide how to react in this situation. The result is that some form of garbage collection is needed to prune the entries. It is RECOMMENDED that when the SEND SAVI device runs out of the memory allocated for the SEND SAVI Data Base, it creates new entries by deleting the entries which Creation Time is higher. This implies that older entries are preserved and newer entries overwrite each

other. In an attack scenario where the attacker sends a batch of data packets with different source address, each new source address is likely to rewrite another source address created by the attack itself. It should be noted that entries are also garbage collected using the LIFETIME, which is updated by NUD_NSOL/NUD_NADV exchange. The result is that in order for an attacker to actually fill the SAVI DB with false source addresses, it needs to continuously answer to NUD_NSOL for all the different source addresses, in order for the entries to grow old and compete with the legitimate entries. The result is that the cost of the attack for the attacker is highly increased.

In addition, it is also RECOMMENDED that a SEND SAVI device reserves a minimum amount of memory for each available port (in the case where the port is used as part of the L2 anchor). The recommended minimum is the memory needed to store 4 bindings associated to the port. The motivation for this recommendation is as follows: an attacker attached to a given port of a SEND SAVI device may attempt to launch a DoS attack towards the SEND SAVI device by creating many bindings for different addresses. It can do so, by sending DAD NSOL for different addresses. The result is that the attack will consume all

the memory available in the SEND SAVI device. The above recommendation aims to reserve a minimum amount of memory per port, so that nodes located in different ports can make use of the reserved memory for their port even if a DoS attack is occurring in a different port.

As the SEND SAVI device may store data packets while the address is being verified, the memory for data packet storage may also be a target of DoS attacks. The effects of such attacks may be limited to the lack of capacity to store new data packets. The effect of such attack will be then that data packets will be dropped during the verification period. A SEND SAVI device MUST limit the amount of memory used to store data packets, allowing the other functions to have available memory even in the case of an attacks as the above described.

It is worth to note that the potential of Denial of Service attacks against the SEND SAVI network is increased due to the use of costly cryptographic operations in order to validate the address of the nodes. An attacker could generate packets using new source addresses in order to make the closest SEND SAVI device spend CPU time to validate DAD_NSOL messages or to generate a secure NUD_NSOL. This attack can be used to drain CPU resources of SEND SAVI devices with a very low cost for the attacker. In order to solve this problem, rate-limiting the processing of packets which may trigger SEND SAVI events SHOULD be enforced in a per-port basis.

5.3. Security Logging

In order to improve the integration of SEND SAVI into an overall security environment, and enable response to additional indirect security issues which SAVI can help ameliorate, it is helpful if SEND SAVI systems log the creation, modification, and deletion of binding entries.

6. IANA Considerations

This document has no actions for IANA.

7. Acknowledgments

Thanks to Ana Kukec for her review and comments on this document. The text has also benefited from feedback provided by Tony Cheneau and Jean-Michel Combes.

Marcelo Bagnulo is partly funded by Trilogy, a research project supported by the European Commission under its Seventh Framework Program.

Alberto Garcia-Martinez is partly funded by T2C2 (TIN2008-06739-C04-01), a Spanish R&D project.

8. References

8.1. Normative References

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

[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.

8.2. Informative References

[RFC2827] 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.

[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-05 (work in progress),
July 2011.

Authors' Addresses

Marcelo Bagnulo
Universidad Carlos III de Madrid
Av. Universidad 30
Leganes, Madrid 28911
SPAIN

Phone: 34 91 6248814
Email: marcelo@it.uc3m.es
URI: http://www.it.uc3m.es

Alberto Garcia-Martinez
Universidad Carlos III de Madrid
Av. Universidad 30
Leganes, Madrid 28911
SPAIN

Phone: 34 91 6248782
Email: alberto@it.uc3m.es
URI: http://www.it.uc3m.es