SAVI Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 4, 2013
April 2, 2013
M. Bagnulo
A. Garcia-Martinez
UC3M

SEND-based Source-Address Validation Implementation
draft-ietf-savi-send-09

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

Copyright Notice

Table of Contents

1. 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 the DAD_NSOL (Duplicate Address Detection Neighbor SOLicitation) and the DAD_NADV (DAD Neighbor ADVertisement) messages defined in [RFC4862], and the NUD_NSOL (Neighbor Unreachability Detection Neigbor SOLicitation) and NUD_NADV (NUD Neighbor ADVertisement) messages defined in [RFC4861] to validate the address ownership claim of a node. In addition, SEND SAVI uses RADV (Router ADVertisement) messages defined in [RFC4861] 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 switch ports, 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 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 because 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 existing 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].

2. Background to SEND SAVI

2.1. 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 called local traffic. Routers may send packets containing a source address other than their own, since they can forward 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. However, 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.

2.2. Binding Creation for SEND SAVI

Filtering is performed according to bindings 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.

Any SAVI solution is not stronger than the binding anchor it uses. If the binding anchor is easily spoofable (e.g., a Media Access Control (MAC) address), then the resulting solution will be weak. The treatment of non-compliant packets needs to be tuned accordingly. In particular, if the binding anchor is easily spoofable and the SEND SAVI device is configured to drop non-compliant packets, then the usage of FCFS SAVI may open a new vector of Denial-of-Service (DoS) attacks, based on spoofed binding anchors. For that reason, in this specification, only switch ports MUST be used as binding anchors. Other forms of binding anchors are out of the scope of this specification, and proper analysis of the implications of using them, should be performed before their usage.

SEND [RFC3971] provides tools to assure that a ND (Neighbor Discovery) message containing a CGA (Cryptographically Generated Addresses) 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 CGA and the port of the SEND SAVI device from which it is reasonable to receive packets with the CGA as source addresses. The events that trigger the binding creation process in a SEND SAVI device are:

- o 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.
- o The reception of any other packet (including data packets) with a source address for which no binding exists. This would occur if DAD_NSOL messages were lost, 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 changed.

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 the case in which the binding creation process initiated by a DAD_NSOL exchange, the SEND SAVI device waits for the reception of a validated DAD_NADV message indicating that other node had configured the address before, or validated DAD_NSOL messages arriving from other locations indicating that another node is trying to configure the same address at the same time. 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 sending those packets 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 secured NUD_NSOL message issued by the SEND SAVI device, which had to be answered by a valid NUD_NADV message by the node for which the binding exists.

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

SEND SAVI needs to be protected against replay attacks, i.e., attacks in which a secured SEND message is replayed by another node. As discussed before, the SEND SAVI specification uses SEND messages to create a binding between the address contained in the message (that must be signed by a node possessing the private key associated to the address) and the port through which the message is received. If an attacker manages to obtain such a message from another node, for example because the message was sent to the all-nodes multicast address or because the attacker has subscribed to the Solicited Node multicast address associated to a remote node, it could replay it preserving the original signature. This may create an illegitimate binding in the SEND SAVI device, or could be used to abort address configuration at other node. While SEND provides some means to limit the impact of the replay of ND messages, the emphasis for SEND anti-replay protection is to limit to a short period of time the validity

of the ND information transmitted in the message, for example, the relationship between an IPv6 address and a layer-2 address. Note that the period must be long enough to assure that the information sent by the legitimate sender is considered valid despite the possible differences in clock synchronization between sender and receiver(s). 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 approximately 2 seconds (more precisely, 2/0.99 seconds). The underlying assumption for SEND security is that even if the message is replayed by another node during this period of time, the information disseminated by ND is still the same. However, allowing a node to replay a SEND message do have impact to SEND SAVI operation, regardless the time elapsed since it was generated, since it can create a new binding in a SEND SAVI device for the port to which an illegitimate node attaches. As can be concluded, the protection provided by SEND may be not enough for SEND SAVI.

SEND SAVI is designed to increase the protection against the replay attacks compared to SEND. First, each node is required to connect to the SEND SAVI topology through a different port to prevent eavesdropping before entering to the SAVI protection perimeter. Then, SEND SAVI bindings are updated only according to messages whose dissemination can be restricted in the SEND SAVI topology without interfering with normal SEND operation. The messages used by SEND SAVI to create bindings are DAD_NSOL messages, for which SEND SAVI limits its propagation to the ports through which a previous binding for the same IPv6 address existed (see Section 3.3.2), and NUD_NADV messages in response to a secured NUD_NSOL sent by the SEND SAVI device only through the tested port. Finally, SEND SAVI filtering rules prevent nodes from replaying messages generated by the SEND SAVI devices themselves. Section 4.1 discusses in more detail the protection provided by SEND SAVI against replay attacks.

2.3. SEND SAVI Protection Perimeter

In order to reduce computing and state requirements in SEND SAVI devices, SEND SAVI devices can be deployed to form a "protection perimeter" [I-D.ietf-savi-framework]. With this deployment strategy, source address validation is performed only when packets enter in the protected realm defined through the protection perimeter. The perimeter is defined by appropriate configuration of the roles of

each port, which can be 'Validating' or 'Trusted':

o Validating ports (VPs) are those in which SEND SAVI filtering and binding creation is performed.

o Trusted ports (TPs) are ports in which limited processing is performed. Only SEND messages related with certificates, prefix information and DAD operation are processed, in order to update the state of the SEND SAVI device or the state related with any of the Validating ports of the switch.

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

```
         +--+   +--+                          +--+   +--+
         |H1|   |H2|                          |H3|   |R1|
         +--+   +--+                          +--+   +--+
           |     |                              |     |
      +----------SEND SAVI PROTECTION PERIMETER-------------+
      |    |     |                              |     |     |
      |  +-1-----2-+                          +-1-----2-+   |
      |  |  SEND-  |                          |  SEND-  |   |
      |  |  SAVI1  |                          |  SAVI2  |   |
      |  +-3--4----+                          +--3--4---+   |
      |    |  |          +--------------+        |  |       |
      |    |  +----------|              |--------+  |       |
      |    |             |   SWITCH-A   |           |       |
      |    |  +----------|              |           |       |
      |    |  |          +--------------+           |       |
      |  +-1--2----+                          +-----1---+   |
      |  |  SEND-  |                          |  SEND-  |   |
      |  |  SAVI3  |                          |  SAVI4  |   |
      |  +-3-----4-+                          +----4----+   |
      |    |     |                                 |        |
      +------------SEND SAVI PROTECTION PERIMETER-----------+
           |     |                                 |
         +--+   +--+                             +--+
         |R2|   |H4|                             |H5|
         +--+   +--+                             +--+
```


Trusted ports are used for connections with trusted infrastructure,

such as other SEND SAVI devices. Port 3 of SEND-SAVI1 and port 1 of SEND-SAVI3, and port 4 of SEND-SAVI2 and port 1 of SEND-SAVI4 are trusted because they connect two SAVI devices. Port 4 of SEND-SAVI1, port 3 of SEND-SAVI2 and port 2 of SEND-SAVI3 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. Therefore, hosts are normally connected to Validating ports. Routers are also recommended to be connected to Validating ports, although they could also be attached to Trusted ports. For a more detailed discussion on this, see Section 3.4. 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 host H5.

2.4. Special cases

Multi-subnet links: In some cases, a given subnet may have several prefixes. This is supported by SEND SAVI as any port can support multiple prefixes.

Multihomed hosts: A multihomed host is a host with multiple interfaces. The interaction between SEND SAVI and multihomed hosts is as follows. If the different interfaces of the host are assigned different IP addresses and packets sent from each interface always carry the address assigned to that interface as source address, then from the perspective of a SEND SAVI device, this is equivalent to two hosts with a single interface, each with an IP address. This is supported by SAVI without need for additional considerations. If the different interfaces share the same IP address or if the interfaces have different addresses but the host sends packets using the address of one of the interfaces through any of the interfaces, then SEND SAVI does not directly support it. It would require either connecting at least one interface of the multihomed host to a Trusted port, or manually configure the SEND SAVI bindings to allow binding the address of the multihomed host to multiple anchors simultaneously.

Untrusted routers: One can envision scenarios where routers are dynamically attached to a SEND SAVI network. A typical example would be a mobile phone connecting to a SEND SAVI switch where the mobile phone is acting as a router for other personal devices that are accessing the network through it. In this case, the router does not seem to directly fall in the category of Trusted infrastructure (as if this was the case, it is likely that all devices would be

trusted), hence it cannot be connected to a trusted port and if it is connected to a Validating port, the SEND SAVI switch would discard all the packets containing an off-link source address coming from that device. As a result, the default recommendation specified in this specification does not support such a scenario.

3. SEND SAVI Specification

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].

3.1. SEND SAVI Data Structures

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

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 is populated with the contents of validated SEND messages. Each entry contains the following information:

- o IPv6 source address
- o Binding anchor: port through which the packet was received
- o Lifetime
- o Status: TENTATIVE_DAD, TENTATIVE_NUD, VALID, TESTING_VP, TESTING_VP'
- o Alternative binding anchor: port from which a DAD_NSOL message or any data packet has been received while a different port was stored in the binding anchor for the address.
- 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 support discovering this information from the Prefix Information option [RFC4861] with the L set bit set of validated RADV messages, either coming from Validating or Trusted ports, as described in Section 3.3.2. The list of prefixes MAY also be configured manually. 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: prefix included in a Prefix Information option
- o Prefix lifetime: time in seconds that the prefix is valid.

Initially set to the Valid Lifetime value of the Prefix Information option of a valid RADV message, or set to a value of

all one bits (0xffffffff), which represents infinity, if configured manually.

When the SEND SAVI device boots, it MUST send a Router Solicitation (RSOL) message, which does not need to be secured if the unspecified address is used (see [RFC3971], sections 5.1.1 and 5.2.1). The SAVI device SHOULD issue a RSOL message 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. A SEND SAVI device MUST support discovering this information from a validated RADV message received from a Validating port, addressed to the all-nodes multicast address or to the IPv6 address of the SEND SAVI device. If the SEND SAVI device uses RADV messages to obtain this information, the SAVI device SHOULD issue a 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. Alternatively, the list of routers MAY be configured manually. The information stored in the table is the following:

- o IPv6 address of the Router. 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 Router lifetime: Lifetime associated with the default router in units of seconds. Initially set to the Router Lifetime of a valid RADV message.

3.2. 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 Certification Path Solicitation (CPS) messages,

- o the SEND SAVI device MUST be configured with a valid CGA address. This CGA address SHOULD be a link-local address, to recover from the following situation: the DAD_NSOL message used by a router

when it configures its link-local address has not been received, so a binding has not been created for the router address. If the port to which the router connects is a Validating port, the SEND SAVI device cannot accept any packet, so no RADV issued by the router will be accepted. Then, the SEND SAVI device may not receive prefix configuration to configure any other address than a link-local. However, if the SEND SAVI device configures a link-local CGA, it can issue a NUD_NSOL to the router, and create the binding according to the process described in Section 3.3.2.

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., in addition to fulfill the requirements stated for regular IPv6 nodes [RFC6434].

- o the SEND SAVI device MUST be configured with at least one trust anchor to validate the Certification Paths that is used to validate router information.
- o the SEND SAVI device 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 3.4. Unconfigured ports MUST be labeled as Validating ports; in this case performance may be degraded, as discussed in [I-D.ietf-savi-framework].

3.3. Traffic Processing

In this section we describe how packets are processed by a SEND SAVI device. Behavior varies depending on if the packet belongs 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 or the unspecified address (local traffic), or not included in the SEND SAVI prefix list (transit traffic).

3.3.1. Transit Traffic Processing

Transit traffic processing occurs as follows:

- o If the transit traffic packet is received through a Trusted port, the data packet is forwarded and no SAVI processing performed.

o If the transit traffic packet is received through a Validating port, the packet is only forwarded if the port through which the packet has been received is associated to the port of an IPv6 address for which an entry in the Router list exists. If transit traffic is received from a Validating port 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 through which the packet was received.

3.3.2. 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 refer to nodes.

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

- o 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 a NUD_NSOL message according to the Neighbor Unreachability Detection procedure described in [RFC4861], addressed to the IPv6 target address, which is the source address of the packet triggering the procedure. This message is secured by SEND as defined in [RFC3971]. The source address used for issuing the NUD_NSOL message is the source address of the SEND SAVI device. The message is sent only through port P.
- o 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 IPv6 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 received from the port associated to the IPv6 address are forwarded, and 'non-forwarding' states, i.e., states in which packets different to the ones used for signaling are not forwarded. VALID, TENTATIVE_DAD, TESTING_VP and TESTING_VP' are forwarding states, and NO_BIND and TENTATIVE_NUD are non-forwarding states.

The SEND SAVI device MUST join the Solicited Node Multicast group for all the addresses whose 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 Multicast Listener Discovery (MLD) messages being sent by the node attached to a Validating port for which a binding for the corresponding address exist, since the node may move and packets sent to that particular Solicited Node Multicast group may no longer be forwarded to the SEND SAVI device.

SEND SAVI devices MUST support the processing of validated Certification Path Advertisement (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 updating these structures 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 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.

In order to determine which traffic is on-link and off-link, the SEND SAVI device MUST support discovery of this information from the Prefix Information option with the L set bit set of validated RADV messages. In this case, at least one router MUST be configured to advertise RADV messages containing a Prefix Information option with the prefixes that the untrusted nodes can use as source addresses, and the bit L set. An alternative to this is to configure manually the SEND SAVI prefix list.

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

- o 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. In other words, 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.

  The reduced set of messages being able to trigger a change simplifies the processing at SEND SAVI devices.
- o DAD_NADV and NUD_NADV are only processed when they are a response to a DAD_NSOL or a NUD_NSOL message.
- o 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 attempt to validate them in order to reduce processing load at the SEND SAVI device.
- o The only messages the SEND SAVI device is required to generate

specifically per each source IP address are MLD and NUD_NSOL messages. This also keeps the state machine simple.

- o 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.
- o If a node has an address configured, and it can prove that it owns 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.

We next describe how different inputs are processed depending on the state of the binding of the IP address 'IPaddr'. Note that every ND message is assumed to be validated according to SEND specification.

To facilitate the reader understanding the most relevant transitions of the SEND SAVI state machine, a simplified version, which does not contain every possible transition, is depicted in the next figure:


```
                  +-------------+
                  |             |
                  | TESTING_VP' |
                  |             |
```

```
                          +-------------+
                             |    ^
            Timeout / VP=VP' |    |
                VP_NUD_NADV  |    |  VP'_DAD_NSOL/
                             |    |    NUD_NSOL
                             |    |
                             v    |
          VP_DAD_NSOL      +--------+
            +------------- |        |
            |              | VALID  |< -------------------+
            |   +-------- >|        |                     |
            |   |          +--------+                     |
            |   |            ^    |                       |
            |   |    VP_NUD_ |    | Timeout,              |
            |   |     NADV/- |    | TP_DAD_NSOL/NUD_NSOL  |
            |   |            |    v                       |
            |   |         +------------+                  |
            |   |         |            |                  |
            |   |         | TESTING_VP |                  |
            |   |         |            |                  |
            |   |         +------------+                  |
            |   |              |                          |
            |   |              | Timeout                  |
            |   | VP*,         |                          |
            |   | Timeout/-    |              VP_NUD_NADV |
            v   |              |                          |
         +---------------+     |           +---------------+
         |               |     |           |               |
         | TENTATIVE_DAD |     |           | TENTATIVE_NUD |
         |               |     |           |               |
         +---------------+     |           +---------------+
            ^  |               |             |         ^
            |  |               |   Timeout/- |         |
            |  | TP_DAD_NSOL,  |             |         |
            |  | TP_DAD_NADV/- |             |         |
            |  |               v             |         |
            |  |           +---------+       |         |
            |  +--------- >|         |< -----+         |
            |              | NO_BIND |                 |
            +--------------|         |-----------------+
            VP_DAD_NSOL/-  +---------+    VP*/VP_NUD_NSOL
```

                Simplified SEND SAVI state machine

NO_BIND

When the node is in this state, there are no unresolved NUD_NSOL messages generated by SEND SAVI or DAD_NSOL propagated to any Validating port, so the only relevant inputs are DAD_NSOL messages coming either from a Validating port (VP) or Trusted port (TP), or any packet other than DAD_NSOL coming from VP or TP. There are no timers configured for this state.

- o 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 LIFETIME 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, sets BINDING_ANCHOR to VP, 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].
- o 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 LIFETIME to TENT_LT. The SEND SAVI device creates a new entry in the SEND SAVI Data Base for IPaddr, sets BINDING_ANCHOR to VP, 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 NUD procedure is being executed, or MAY store it in order to send it if the next transitions are (strictly) TENTATIVE_NUD and then VALID.
- o 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 MUST NOT be forwarded through any of the Validating ports but it is 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.
- o 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 the BINDING_ANCHOR port and it has forwarded it to the appropriate TPs. The relevant events occurring in this state are: the reception of a DAD_NADV message from a TP, a DAD_NSOL message from the BINDING_ANCHOR port, other Validating port or TP, a data packet from the BINDING_ANCHOR port, and the expiration of the LIFETIME timer initiated when the DAD_NSOL was received at port the BINDING_ANCHOR port.

o 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 the BINDING_ANCHOR port. The state is changed to NO_BIND, and the LIFETIME cleared.
o 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 the BINDING_ANCHOR port, so that the node at this port will not configure the address, as stated in [RFC4862]. The DAD_NSOL message is also forwarded to all appropriate Trusted ports. Then, the LIFETIME is cleared, and the state is changed to NO_BIND.
o 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.
o If a DAD_NSOL is received from a Validating port VP' different the BINDING_ANCHOR port, 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 the BINDING_ANCHOR port, so that the node at this port will not configure the address, as stated in [RFC4862]. The DAD_NSOL message is also forwarded to all appropriate Trusted ports. Then, the BINDING_ANCHOR is set to VP' (through which the DAD_NSOL message was received), the LIFETIME is set to TENT_LT, and the state

remains in TENTATIVE_DAD.

o Any other packet than a validated DAD_NSOL is received from a Validating port VP' different from the BINDING_ANCHOR port is discarded. The state is not changed.

o If a DAD_NSOL is received from the BINDING_ANCHOR port, 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 LIFETIME is set to TENT_LT, and the state remains in

TENTATIVE_DAD.

o If any packet other than a DAD_NSOL is received from the BINDING_ANCHOR port, 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 LIFETIME 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 LIFETIME is not changed, and the state remains 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 them) RADV messages addressed to destinations other than the all-nodes multicast address or to the IPv6 address of the SEND SAVI device.

o If LIFETIME expires, it is assumed that no other node has configured this address. Therefore, the Validating port VP (currently stored in the BINDING_ANCHOR) could be bound to this IPv6 address. The LIFETIME 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 the BINDING_ANCHOR port, other Validating port or a TP, and any packet other than DAD_NSOL from other validating port than

the BINDING_ANCHOR or a TP. The expiration of LIFETIME is also relevant to trigger a check for address ownership for the node at the BINDING_ANCHOR port.

- o If a DAD_NSOL with IPaddr as source address is received through the BINDING_ANCHOR port, 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 LIFETIME 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 the BINDING_ANCHOR port 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.
- o 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 LIFETIME is set to TENT_LT, a secured NUD_NSOL message is sent to IPaddr through VP and the state is changed to TESTING_VP.
- o 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 the BINDING_ANCHOR port, the LIFETIME is set to TENT_LT, and the state is changed to TESTING_VP.
- o If a DAD_NSOL packet with IPaddr as source address is received through a Validating Port VP' (VP' different from the current BINDING ANCHOR), 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 the BINDING_ANCHOR port. In addition, a secured NUD_NSOL is sent to BINDING_ANCHOR port, the ALTERNATIVE BINDING ANCHOR is set to port VP' (for future use if the node at VP' is finally selected), the LIFETIME is set to

TENT_LT, and the state is changed to TESTING_VP'.

- o If any packet other than a DAD_NSOL with IPaddr as source address is received from a Validating port VP', different from the current BINDING_ANCHOR for this binding, VP, the packet is discarded. The SEND SAVI device MAY issue a secured NUD_NSOL through the BINDING_ANCHOR port, store VP' in the ALTERNATIVE BINDING ANCHOR for possible future use, set the LIFETIME 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'.
- o If LIFETIME expires, a secured NUD_NSOL message is sent through the BINDING_ANCHOR port to IPaddr, the LIFETIME 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.

- o 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 LIFETIME is changed to DEFAULT_LT, and the state is changed to VALID. The message is not forwarded to any other port.
- o 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 LIFETIME is set to DEFAULT_LT, and the state is changed to TENTATIVE_DAD.

o 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.

o Any packet other than DAD_NSOL or NUD_NADV containing IPaddr as a source address arriving from the BINDING_ANCHOR port is forwarded. Neither the LIFETIME nor the state are changed.

o 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 LIFETIME nor the state are changed. The node at the BINDING_ANCHOR port is under check: if it still is at this port, 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 and the local state will move to NO_BIND.

o If a packet other than a DAD_NSOL arrives from a Trusted port, the packet is forwarded. Neither the LIFETIME nor the state are changed.

o If a DAD_NSOL is received from a Validating port VP' other than the current BINDING_ANCHOR port, 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 the BINDING_ANCHOR port and to the appropriate Trusted ports. In addition, a secured NUD_NSOL is sent to the BINDING_ANCHOR port, the ALTERNATIVE BINDING ANCHOR is set to VP' (for future use if the node at VP' is finally selected), the LIFETIME is set to TENT_LT, and the state is changed to TESTING_VP'.

o Any other packet received from a Validating port VP' other than the BINDING_ANCHOR port 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'.

- o If the LIFETIME expires, the LIFETIME is cleared and the state is changed to NO_BIND.

TESTING_VP'

To arrive to this state an indication that a node at VP' different from the BINDING_ANCHOR port wants to send data with IPaddr as source address occurred while a binding existed for VP. The port VP' which triggered the change of the state to TESTING_VP' was stored at the ALTERNATIVE_BINDING_ACNHOR, 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 the BINDING_ANCHOR port. The relevant events that may occur in this case are the reception of a NUD_NADV from port VP (the BINDING_ANCHOR port), 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.

- o If a NUD_NADV is received from the BINDING_ANCHOR port, 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, the LIFETIME is set to DEFAULT_LT, and the state is changed to VALID.
- o If a DAD_NSOL is received from the BINDING_ANCHOR port, 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 port stored in the ALTERNATIVE_BINDING_ANCHOR). The BINDING_ANCHOR in use is kept, the LIFETIME 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.

- o If a RADV message is received from the BINDING_ANCHOR port, 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 LIFETIME 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.

- o Any packet other than a validated DAD_NSOL, a validated NUD_NADV or a validated RADV coming from the BINDING_ANCHOR port, is forwarded, and the state is not changed.
- o If a DAD_NSOL is received from the port stored in the ALTERNATIVE_BINDING_ANCHOR, 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 BINDING_ANCHOR port. The BINDING_ANCHOR and the ALTERNATIVE BINDING ANCHOR are kept, the LIFETIME is set to DEFAULT_LT, and the state is not changed.
- o Any packet other than a DAD_NSOL coming from the ALTERNATIVE_BINDING_ANCHOR port is discarded, and the state is not changed.
- o If a DAD_NSOL is received from port VP", different from BINDING_ANCHOR and the ALTERNATIVE_BINDING_ANCHOR ports, 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 BINDING_ANCHOR and the ALTERNATIVE_BINDING_ANCHOR ports. The node at ALTERNATIVE BINDING ANCHOR port is expected to unconfigure its address if the message triggering the transition to this state was a DAD_NSOL message received from the ALTERNATIVE_BINDING_ANCHOR port (and not any other packet). The state remains in TESTING_VP' although VP" is stored in the ALTERNATIVE_BINDING_ANCHOR for future use if the node at VP" is finally selected. The LIFETIME is not changed.
- o Any packet other than a DAD_NSOL received from port VP" is discarded and does not affect to the state.
- o 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 the BINDING_ANCHOR and the ALTERNATIVE_BINDING_ANCHOR ports and other appropriate Trusted ports. The LIFETIME is left unchanged and the state is changed to TESTING_VP. The node at the ALTERNATIVE_BINDING_ANCHOR port is expected to unconfigure its address if the packet triggering the transition to this state was a DAD_NSOL message received from the

ALTERNATIVE_BINDING_ANCHOR port.

- o Any packet other than a DAD_NSOL coming from a Trusted port is forwarded appropriately, but the state is not changed.
- o If LIFETIME expires, it is assumed that the node for which the binding existed is no longer connected through the BINDING_ANCHOR port. Therefore, the BINDING_ANCHOR is set to the ALTERNATIVE_BINDING_ANCHOR port value. The LIFETIME 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 the BINDING_ANCHOR port without any existing binding in the SEND SAVI device. The SEND SAVI device has sent a NUD_NSOL message to the BINDING_ANCHOR port. The relevant events for this case are the reception of a NUD_NADV from port the BINDING_ANCHOR port; the reception of DAD_NSOL from the BINDING_ANCHOR port, other VP different from the BINDING_ANCHOR port, or a TP; and the reception of any packet other than DAD_NSOL and NUD_NADV from the BINDING_ANCHOR port, and other than DAD_NSOL for other VP different than the BINDING_ANCHOR port, or TP. In addition, the LIFETIME may expire.

- o If a NUD_NADV message is received through the BINDING_ANCHOR port, the SEND SAVI device checks for its validity. If the message is not valid, it MUST be discarded. If the message is valid, the LIFETIME is set to TENT_LT, and the state is changed to VALID. The message is not forwarded to any port.
- o If a DAD_NSOL message is received through the BINDING_ANCHOR port, 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 LIFETIME is set to TENT_LT and the state is changed to TENTATIVE_DAD.
- o Any packet other than NUD_NADV or DAD_NSOL received through the BINDING_ANCHOR port is discarded.
- o If a DAD_NSOL message is received through port VP' different from the BINDING_ANCHOR port, 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 LIFETIME is set to TENT_LT, the BINDING_ANCHOR is set to VP', and the state is changed to TENTATIVE_DAD.
- o 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.
- o 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 the BINDING_ANCHOR port, and the state is left unchanged.

- o 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.
- o If LIFETIME expires, the LIFETIME is cleared and the state is changed to NO_BIND.

3.4. SEND SAVI Port Configuration Guidelines

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

- o 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.
- o 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.
- o No more than one host SHOULD be connected to each port. Not doing so will allow hosts to generate packets with the same source address as the other hosts connected to the same port, and will allow performing replaying attacks as described in Section 4.1.
- o 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 ports, 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.
- o 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.

3.5. VLAN Support

In the case the SEND SAVI device is a switch that supports customer VLANs [IEEE.802-1Q.2005], the SEND SAVI implementation MUST behave as if there was one SEND SAVI process per customer VLAN. The SEND SAVI process of each customer VLAN will store the binding information

corresponding the nodes attached to that particular customer VLAN.

3.6. Protocol Constants

TENT_LT is 500 milliseconds.

DEFAULT_LT is 5 minutes.

4. Security Considerations

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.

4.1. 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 SEND messages used by the SEND SAVI specification. An attacker could replay any of these messages to interfere with SEND SAVI operation. For example, it could replay a DAD_NSOL message to abort the configuration of an address for a legitimate node and to gain the right to use the address for DEFAULT_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 SEND SAVI reply attacks, we have to consider two different cases:

- When the SEND message replayed is used to create or update binding information for SEND SAVI, since the port through which this message is received is key to SEND SAVI operation. 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 SEND message replayed does not result in the update of binding information for SEND SAVI, and thus it is not related to the specific port through which it was received. Such situations are the reception of CPA messages containing certificates, and 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 two cases, the security risks are equivalent to those of SEND operation, i.e., we can consider that the information will not be changed by its legitimate sender for the time during which the SEND specification allows replaying (which depends on the values of TIMESTAMP_FUZZ and TIMESTAMP_DRIFT, [RFC3971]). 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 binding, the replay of this RADV message does not provide an advantage to an attacker. This is so because SEND SAVI requires a binding to exist (between the IPv6 address and the port of the SEND SAVI device) prior to consider the RADV message, so protecting the creation of the binding also protects the ability of an attacker to become a router.

We now discuss the protection provided by SEND SAVI against the

replay of messages used to create or update bindig information, i.e., the replay of DAD_NSOL and NUD_NSOL/NUD_NADV messages. In this case, protection results from requiring a one-to-one correspondence between SEND SAVI ports and nodes connected to the link (see Section 3.4), and careful filtering when transmitting the messages involved in SEND SAVI operation. Note that if many nodes are attached to the same SEND SAVI Validating port, i.e., the one-to-one correspondence is violated, any of them can generate packets with the legitimate source address of any other node, defeating the source address validation of SEND SAVI. Moreover, any of these nodes may interfere with the communication capacity of the legitimate node in many ways, as it is considered next. Assume two nodes H1 and H2 are connected to switch A, not enabled for SEND SAVI operation, which accesses to the SEND SAVI protection perimeter through port 1. H1 is switched off. Node H2 knows IP1, the address H1 will configure when it switches on, so H2 subscribes to the Solicited Node of IP1 address. Although H2 cannot generate a valid SEND message for H1's address, when H1 switches on H2 receives the DAD_NSOL issued by H1, and replays it in a time shorter than the time required to invalidate the SEND message. When H1 receives a valid DAD_NSOL message for its own address, it

stops its address configuration process for IP1. The SEND SAVI device receives this second message, but it has no way to know that the message has been issued by a different node, so it forwards it. After TENT_LT time, the binding is configured in the SEND SAVI device, and H2 can use IP1 for DEFAULT_LT time. Alternatively, the SEND SAVI binding could also be configured in a different port, provided that there exists a host H3 connected to that port which receives from H2 (using a tunnel, to prevent the processing from a SEND SAVI device) the DAD_NSOL message legitimately issued by H1.

```
                 +----------+            |
                 |  SEND    |            | +--+
                 |  SAVI    2--------------|H3|
                 +----1-----+            | +--+
                      |                  |
      ----SEND SAVI PROTECTION PERIMETER---+
                      |
                 +----------+
                 |SWITCH A  |
```

```
                    +---------+
                     |       |
                    +--+   +--+
                    |H1|   |H2|
                    +--+   +--+
```

If a one-to-one correspondence among ports and hosts is honored, the traffic generated by a node cannot be captured before arriving to the SEND SAVI protection perimeter. In this case, the protection provided by SEND SAVI is the following:

- o To prevent the replay of DAD_NSOL messages, SEND SAVI devices only forward them to ports for which a binding to the address being tested by the DAD_NSOL message existed. Therefore, it is not enough for an attacker to subscribe to a Solicited Node address to receive DAD_NSOL messages sent to that address, but the attacker needs to generate a valid DAD_NSOL message associated to the address for which the binding is being tested, which is deemed unfeasible [RFC3971].

4.2. Protection Against Denial of Service Attacks

The attacks against the SEND SAVI device basically consist of making the SEND SAVI device consume its resources until it runs out of them. For instance, a possible attack would be to send packets with different source addresses, making the SEND SAVI device create state for each of the addresses and waste memory. At some point, the SEND

SAVI device runs out of memory and needs to decide how to react. The result is that some form of garbage collection is needed to prune the entries. When the SEND SAVI device runs out of the memory allocated for the SEND SAVI Data Base, it is RECOMMENDED that it create new entries by deleting the entries with a higher Creation time. 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 addresses, 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 DEFAULT_LT, which is updated by NUD_NSOL/NUD_NADV exchange. The result is that in order for an attacker to actually fill the FCFS SAVI Data Base with false source addresses, it needs to continuously answer to NUD_NSOL for all the different source

addresses so that the entries grow old and compete with the legitimate entries. The result is that the cost of the attack is highly increased for the attacker.

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 such those described above.

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.

4.3. Residual threats

SEND SAVI assumes that a host will be able to defend its address when the DAD procedure is executed for its addresses, and that it will answer to a NUD_NSOL with a NUD_NADV when required. This is needed, among other things, to support mobility within a link (i.e., to allow a host to detach and reconnect to a different Layer_2 anchor of the same IP subnetwork, without changing its IP address). If the SEND SAVI device does not see the DAD_NADV or the NUD_NADV, it may grant the binding to a different binding anchor. This means that if an attacker manages to prevent a host from defending its source address, it will be able to destroy the existing binding and create a new one, with a different binding anchor. An attacker may do so for example by launching a DoS attack to the host that will prevent it to issue proper replies.

4.4. Privacy considerations

Personally identifying information MUST NOT be included in the SEND SAVI Data Base with the MAC address as the canonical example, except when there is an attempt of attack involved. Moreover, compliant implementation MUST NOT log binding anchor information except where there is an identified reason why that information is likely to be involved in detection, prevention or tracing of actual source address spoofing. Information that is not logged MUST be deleted as soon as possible (i.e., as soon as the state for a given address is back to NO_BIND). Information about the majority of hosts that never spoof SHOULD NOT be logged.

5. IANA Considerations

This document has no actions for IANA.

6. Acknowledgments

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

Marcelo Bagnulo is partly funded by Trilogy, a research project

supported by the European Commission under its Seventh Framework Program.

7. References

7.1. 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.

7.2. 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-06 (work in progress), January 2012.

[RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", RFC 6434, December 2011.

[IEEE.802-1Q.2005]
Institute of Electrical and Electronics Engineers, "IEEE Standard for Local and metropolitan area networks / Virtual Bridged Local Area Networks", IEEE Standard 802.1Q, May 2005.

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