Workgroup:
IPv6 Maintenance (6man) Working Group
Internet-Draft: draft-ietf-6man-slaac-renum-05
Updates: 4191, 4861, 4862, 8106 (if approved)
Published: 13 March 2023
Intended Status: Standards Track
Expires: 14 September 2023
Authors: F. Gont   J. Zorz   R. Patterson
         SI6 Networks   6connect   Sky UK

Improving the Robustness of Stateless Address Autoconfiguration (SLAAC) to Flash Renumbering Events

Abstract

In renumbering scenarios where an IPv6 prefix suddenly becomes invalid, hosts on the local network will continue using stale prefixes for an unacceptably long period of time, thus resulting in connectivity problems. This document improves the reaction of IPv6 Stateless Address Autoconfiguration to such renumbering scenarios.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In scenarios where network configuration information becomes invalid without any explicit signaling of that condition, hosts on the local network will continue using stale information for an unacceptably long period of time, thus resulting in connectivity problems. This problem has been discussed in detail in [RFC8978].

This document updates the Neighbor Discovery specification [RFC4861], the Stateless Address Autoconfiguration (SLAAC) specification [RFC4862], and other associated specifications ([RFC4191] and [RFC8106]), such that hosts can more gracefully deal with the so-called flush renumbering events, thus improving the robustness of SLAAC.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. SLAAC reaction to Flash-renumbering Events

In some scenarios, the local router triggering the network renumbering event may try to deprecate the stale information (by explicitly signaling the network about the renumbering event), whereas in other scenarios the renumbering event may happen inadvertently, without the router explicitly signaling the scenario to local hosts. The following subsections analyze specific considerations for each of these scenarios.

3.1. Renumbering without Explicit Signaling

In the absence of explicit signalling from SLAAC routers (such as sending Prefix Information Options (PIOs) with small lifetimes to deprecate stale prefixes), stale prefixes will remain preferred and valid according to the Preferred Lifetime and Valid Lifetime parameters (respectively) of the last received PIO. [RFC4861] specifies the following default values for PIOs:

* Preferred Lifetime (AdvPreferredLifetime): 604800 seconds (7 days)

* Valid Lifetime (AdvValidLifetime): 2592000 seconds (30 days)

This means that, in the absence of explicit signaling by a SLAAC router to deprecate a prefix, it will take a host 7 days (one week) to deprecate the corresponding addresses, and 30 days (one month) to eventually remove any addresses configured for the stale prefix. Clearly, employing such long default values is generally unacceptable for most deployment scenarios that may experience flash-renumbering events.

NOTE:

[RFC8978] provides an operational recommendation for Customer Edge (CE) routers to override the standard default Preferred

Lifetime (AdvPreferredLifetime) and Valid Lifetime (AdvValidLifetime) to 2700 seconds (45 minutes) and 5400 seconds (90 minutes), respectively, thus improving the state of affairs for CE router scenarios.

Similarly, other Neighbor Discovery optons may employ unnecessarily long lifetimes that may be unacceptable for most deployment scenarios that may experience flash-renumbering events.

Use of more appropriate timers in Router Advertisement messages can help limit the amount of time that hosts will maintain stale configuration information. Additionally, hosts may normally in a position to infer network configuration has changed -- for example, if a router ceases to advertise previously-advertised information.

Section 4.1 formally specifies the use of more appropriate (i.e., shorter) default lifetimes for Neighbor Discovery options, while Section 4.5 specifies a local policy that SLAAC hosts may implement to infer that network configuration information has changed, such that stale configuration information can be phased out.

3.2. Renumbering with Explicit Signaling

In scenarios where a local router is aware about the renumbering event, it may try to phase out the stale network configuration information. In these scenarios, there are two aspects to be considered:

* The amount of time during which the router should continue trying to deprecate the stale network configuration information

* The ability of SLAAC hosts to phase out stale configuration in a timelier manner.

Since the network could be become partitioned at any arbitrary time and for an arbitrarily long period of time, routers need to contemplate the possible scenario where hosts receive an RA message, and the network subsequently becomes partitioned. This means that in order to reliably deprecate stale information, a router would should try to deprecate it for a period of time equal to the associated Neighbor Discovery option lifetime used when advertising the information.

NOTE:

For example, it should try to deprecate a prefix (via a PIO) for a period of time equal to the "Preferred Lifetime" used when advertising the prefix, and try to invalidate the prefix for a period of time equal to the "Valid Lifetime" (see Section 12 of [RFC4861]) used when advertising the prefix.

Once the number of seconds in the original "Preferred Lifetime" have elapsed, all hosts would have deprecated the corresponding addresses anyway, while once the number of seconds in the "Valid Lifetime" have elapsed, the corresponding addresses would be invalidated and removed.

Thus, use of more appropriate default lifetimes for Neighor Discovery options, as specified in Section 4.1, would reduce the amount of time stale options would need to be announced as such by a router in order to ensure that it is deprecated/invalidated.

In the case of Prefix Information Options (PIOs), in scenarios where a router has positive knowledge that a prefix has become invalid and thus could signal this condition to local hosts, the current specifications will prevent SLAAC hosts from fully recovering from such stale information: Item "e)" of Section 5.5.3 of [RFC4862] specifies that an RA may never reduce the "RemainingLifetime" to less than two hours. Additionally, if the RemainingLifetime of an address is smaller than 2 hours, then a Valid Lifetime smaller than 2 hours will be ignored. The inability to invalidate a stale prefix may prevent communications with the new "owners" of a prefix, and thus is highly undesirable. On the other hand, the Preferred Lifetime of an address *may* be reduced to any value to avoid the use of a stale prefix for new communications.

Section 4.2 formally updates [RFC4862] to remove this restriction, such that hosts may react to the advertised "Valid Lifetime" even if it is smaller than 2 hours. Section 4.3 recommends that routers disseminate network configuration information when a network interface is initialized, such that new configuration information propagates in a timelier manner.

4. Improvements to Stateless Address Autoconfiguration (SLAAC)

The following subsections update [RFC4861] and [RFC4862], such that the problem discussed in this document is mitigated. The updates in the following subsections are mostly orthogonal, and mitigate different aspects of SLAAC that prevent a timely reaction to flash renumbering events.

* Reduce the default Valid Lifetime and Preferred Lifetime of PIOs (Section 4.1):

  This helps limit the amount of time a host may employ stale information, and also limits the amount of time a router needs to try to deprecate stale information.

*Honor PIOs with small Valid Lifetimes (Section 4.2):

This allows routers to invalidate stale prefixes, since otherwise [RFC4861] would prevent hosts from honoring PIOs with a Valid Lifetime smaller than two hours.

*Recommend routers to retransmit configuration information upon interface initialization/reinitialization (Section 4.3):

This helps spread the new information in a timelier manner, and also deprecate stale information via host-side heuristics (see Section 4.5).

*Recommend routers to always send all options (i.e. the complete configuration information) in RA messages, and in the smallest possible number of packets (Section 4.4):

This helps propagate the same information to all hosts, and also allows hosts to better infer that information missing in RA messages has become stale (see Section 4.5).

*Infer stale network configuration information from received RAs (Section 4.5):

This allows hosts to deprecate stale network configuration information, even in the absence of explicit signaling.

4.1. More Appropriate Neighbor Discovery Option Lifetimes

This document defines the following variables to be employed for the default lifetimes of Neighbor Discovery options:

*ND_DEFAULT_PREFERRED_LIFETIME: max(AdvDefaultLifetime, 3 * MaxRtrAdvInterval)

*ND_DEFAULT_VALID_LIFETIME: 2 * ND_DEFAULT_PREFERRED_LIFETIME

where:

AdvDefaultLifetime:

Router configuration variable specified in [RFC4861], which specifies the value to be placed in the Router Lifetime field of Router Advertisements sent from the interface, in seconds.

MaxRtrAdvInterval:

Router configuration variable specified in [RFC4861], which specifies the maximum time allowed between sending unsolicited multicast Router Advertisements from the interface, in seconds.

max():

A function that computes the maximum of its arguments.

NOTE:

The expression above computes of maximum among AdvDefaultLifetime and "3 * MaxRtrAdvInterval" (the default value of AdvDefaultLifetime, as per [RFC4861]) to accommodate the case where an operator might simply want to disable one local router for maintenance, while still having the router advertise SLAAC configuration information.

[RFC4861] specifies the default value of MaxRtrAdvInterval as 600 seconds, and the default value of AdvDefaultLifetime as 3 * MaxRtrAdvInterval. Therefore, when employing default values for MaxRtrAdvInterval and AdvDefaultLifetime, the default values of ND_DEFAULT_PREFERRED_LIFETIME and ND_DEFAULT_VALID_LIFETIME become 1800 seconds (30 minutes) and 3600 seconds (1 one hour), respectively. We note that when implementing BCP202 [RFC7772], AdvDefaultLifetime will typically be in the range of 45-90 minutes, and therefore the value of ND_DEFAULT_PREFERRED_LIFETIME will be in the range 45-90 minutes, while the value of ND_DEFAULT_VALID_LIFETIME will be in the range of 90-180 minutes.

This document formally updates [RFC4861] to modify the default values of the Preferred Lifetime and the Valid Lifetime of PIOs as follows:

* AdvPreferredLifetime: ND_DEFAULT_PREFERRED_LIFETIME

* AdvValidLifetime: ND_DEFAULT_VALID_LIFETIME

This document formally updates [RFC4191] to specify the default Route Lifetime of Route Information Options (RIOs) as follows:

* Route Lifetime: Default: ND_DEFAULT_PREFERRED_LIFETIME

This document formally updates [RFC8106] to modify the default Lifetime of Recursive DNS Server Options as:

* Lifetime: Default: ND_DEFAULT_PREFERRED_LIFETIME

Additionally, this document formally updates [RFC8106] to modify the default Lifetime of DNS Search List Options as:

*Lifetime: Default: ND_DEFAULT_PREFERRED_LIFETIME

4.2. Honor Small PIO Valid Lifetimes

The entire item "e)" (pp. 19-20) from Section 5.5.3 of [RFC4862] is replaced with the following text:

e) If the advertised prefix is equal to the prefix of an address configured by stateless autoconfiguration in the list, the valid lifetime and the preferred lifetime of the address should be updated by processing the Valid Lifetime and the Preferred Lifetime (respectively) in the received advertisement.

RATIONALE: This change allows hosts to react to the signal provided by a router that has positive knowledge that a prefix has become invalid.

*The behavior described in [RFC4862] had been incorporated during the revision of the original IPv6 Stateless Address Autoconfiguration specification ([RFC1971]). At the time, the IPNG working group decided to mitigate the attack vector represented by Prefix Information Options with very short lifetimes, on the premise that these packets represented a bigger risk than other ND-based attack vectors [IPNG-minutes].

While reconsidering the trade-offs represented by such decision, we conclude that the drawbacks of the aforementioned mitigation outweigh the possible benefits.

In scenarios where RA-based attacks are of concern, proper mitigations such as RA-Guard [RFC6105] [RFC7113] or SEND [RFC3971] should be implemented.

4.3. Interface Initialization

When an interface is initialized, it is paramount that network configuration information is spread on the corresponding network (particularly in scenarios where an interface has been re-initialized, and the conveyed information has changed). Thus, this document replaces the following text from Section 6.2.4 of [RFC4861]:

In such cases, the router MAY transmit up to MAX_INITIAL_RTR_ADVERTISEMENTS unsolicited advertisements, using the same rules as when an interface becomes an advertising interface.

with:

In such cases, the router SHOULD transmit MAX_INITIAL_RTR_ADVERTISEMENTS unsolicited advertisements, using the same rules as when an interface becomes an advertising interface.

RATIONALE: Use of stale information can lead to interoperability problems. Therefore, it is important that new configuration information propagates in a timelier manner to all hosts.

NOTE:
[RFC9096] specifies recommendations for CPE routers to signal any stale network configuration information.

4.4. Conveying Information in Router Advertisement (RA) Messages

Intentionally omitting information in Router Advertisements may prevent the propagation of such information, and may represent a challenge for hosts that need to infer whether they have received a complete set of SLAAC configuration information. As a result, this section recommends that, to the extent that is possible, RA messages contain a complete set of SLAAC information.

This document replaces the following text from Section 6.2.3 of [RFC4861]:

A router MAY choose not to include some or all options when sending unsolicited Router Advertisements. For example, if prefix lifetimes are much longer than AdvDefaultLifetime, including them every few advertisements may be sufficient. However, when responding to a Router Solicitation or while sending the first few initial unsolicited advertisements, a router SHOULD include all options so that all information (e.g., prefixes) is propagated quickly during system initialization.

If including all options causes the size of an advertisement to exceed the link MTU, multiple advertisements can be sent, each containing a subset of the options.

with:

When sending Router Advertisements, a router SHOULD include all options.

If including all options would cause the size of an advertisement to exceed the link MTU, multiple advertisements can be sent, each containing a subset of the options. In all cases, routers SHOULD convey all information using the smallest possible number of packets, and convey options of the same type in the same packet to the extent possible.

RATIONALE: Sending information in the smallest possible number of packets was somewhat already implied by the original text in [RFC4861]. Including all options when sending RAs leads to simpler code (as opposed to dealing with special cases where specific information is intentionally omitted), and also helps hosts infer when they have received a complete set of SLAAC configuration information. Note that while [RFC4861] allowed some RAs to omit some options, to the best of the authors' knowledge, all SLAAC router implementations always send all options in the smallest possible number of packets. Therefore, this section simply aligns the protocol specifications with existing implementation practice.

4.5. Recovery from Stale Configuration Information without Explicit Signaling

This section specifies an algorithm, "Lifetime Avoidance Algorithm" (LTA), that allows hosts to infer that previously-advertised configuration information (such as autoconfiguration prefixes) has become stale, such that the stale information can be deprecated in a timelier manner. Most of the value of this algorithm is in being able to mitigate the problem discussed in [RFC8978] at hosts themselves, without relying on changes in SLAAC router implementations.

The algorithm consists of two conceptual building-blocks:

* Detection of possible configuration change

* Validation/Refresh of configuration information

Possible configuration changes can be inferred when a SLAAC router (as identified by its link-local address) ceases to advertise a previously-advertised information. Therefore, hosts can record what configuration information has been advertised by each local router, and infer a configuration change when a router ceases to advertise previously-advertises configuration information.

Inscenarios where possible configuration changes have been detected, hosts should poll the local router via unicasted Router Solicitations (RS) to verify that the router in question has indeed ceased to advertise the aforementioned information. If this condition is confirmed, the corresponding configuration information should be discarded.

In the context of multi-prefix/multi-router networks [RFC8028] [RFC8504], SLAAC configuration information should be associated with each advertising router. Thus, when a router ceases to advertise some configuration information:

* If this was the only router advertising the aforementioned information, the information should be discarded.

* If other routers were advertising the aforementioned information, it should simply be dis-associated with the router that ceased to advertise it, and the fate of this information (and configured resources) should depend solely on the routers that continue advertising it.

Implementation of this kind of heuristic allows a timelier reaction to network configuration changes even in scenarios where there is no explicit signaling from the network, thus improving robustness.

As discussed in Section 4.4, [RFC4861] does not require routers to convey all RA options in the same message. Therefore, the algorithm specified in this section is designed such that it can cope with this corner case that, while not found in the deployed Internet, is allowed by [RFC4861].

4.5.1. Target Neighbor Discovery Options

The LTA algorithm SHOULD be applied to the following Neighbor Discovery options:

* Prefix Information Option [RFC4861]

* Route Information Option (RIO) [RFC4191]

* DNS Search Options (RDNSSO) [RFC8106]

* DNS Search List Options (DNSSLO) [RFC8106]

4.5.2. Local State Information and Configuration Variables

In the context of multi-prefix/multi-router networks [RFC8028] [RFC8504], each option from Section 4.5.1 is associated with each advertising SLAAC router. Therefore, hosts should record what configuration information has been advertised by each local router.

NOTE:
Throughout this specification, each router is identified by its link-local address.

Additionally, hosts associate with piece of configuration information received via SLAAC options a timestamp (INFO_LAST variable below) that records the time at which this information was last advertised by a particular router.

NOTE:
While not strictly required, we note that existing implementations may already record a timestamp representing when a piece of information was advertised by a given router as a possible implementation approach to be able to compute the remaining lifetime of that piece of information.

The algorithm specified in this document employs the following variables:

LTA_MODE:
A boolean variable associated with each SLAAC advertising router that specifies whether the local host is currently performing the LTA algorithm for that router. It is initialized to FALSE.

LTA_LAST:
A variable associated with each SLAAC advertising router that stores the time (in seconds) when the local host last entered the LTA algorithm for this router. It is initialized to 0.

RS_LAST:
A variable associated with each SLAAC advertising router that stores the time (in seconds) when the local host last sent a unicasted Router Solicitation to the router in question. It is initialized to 0.

RS_COUNT:
A variable associated with each SLAAC advertising router that stores the number of unicasted Router Solicitations that have been sent to the corresponding router since the last time the LTA algorithm was executed. It is initialized to 0.

RS_COUNT_MAX:
A configuration variable specifying the maximum number of unicasted Router Solicitations that a host will send to a SLAAC advertising router as part of the LTA algorithm. It defaults to 1.

RS_RNDTIME:
A host-wide variable specifying a random amount of time that the host should wait before sending the first unicasted Router Solicitation message to a SLAAC router as part of the LTA

algorithm. It should be initialized to a value in the range from 0 to 5 seconds when the system is bootstrapped.

RS_TIMEOUT:
A host-wide variable specifying the amount of time to wait for a response to a unicasted Router Solicitation sent as part of the LTA algorithm. It defaults to 3 seconds.

INFO_LAST:
A timestamp associated with each piece of SLAAC information (from Section 4.5.1) received from each SLAAC advertising router.

NOTE:
In most cases (e.g., Prefix Information Options and Route Information Options) each neighbor discovery option carries one atomic piece of SLAAC information. In other cases (notably Recursive DNS Server Option [RFC8106] and DNS Search List Option [RFC8106]), a single neighbor discovery option carries multiple atomic pieces of information (i.e., a host might want to prune some recursive DNS server addresses, but not others). This is why this document refers to "piece of SLAAC information" rather than "Negihbor Discovery option" (since one option might carry multiple pieces of information).

RA_WIN:
A host-wide configuration variable specifying a time window over which a SLAAC advertising router may convey all SLAAC configuration information. It is meant to cope with the theoretical case where a router may spread SLAAC information over several RA messages. It defaults to 3 seconds.

LTA_CYCLE:
This variable accounts for the maximum time that may elapse for the entire LTA algorithm to complete. Its value is computed as: LTA_CYCLE=RA_WIN+RS_RNDTIME+RS_COUNT_MAX*RS_TIMEOUT.

4.5.3. Algorithm Specification

Initialization when a new SLAAC advertising router is learned:

```
LTA_MODE=FALSE
LTA_LAST=0
RS_LAST=0
RS_COUNT=0
LTA_CYCLE=RA_WIN+RS_RNDTIME+RS_COUNT_MAX*RS_TIMEOUT
```

Upon receipt of a Router Advertisement message, and after normal processing of the message, perform the following actions:

```
 TIME= time()

 For each piece of SLAAC configuration information advertised by this
     INFO_LAST= TIME


 IF LTA_MODE==FALSE && TIME > (LTA_LAST+LTA_CYCLE)
     IF this RA is missing any previously-advertised information:
         LTA_MODE=TRUE
         LTA_LAST=TIME
```

RATIONALE:

The goal of checking "(LTA_LAST+LTA_CYCLE)" is to prevent the host from re-entering the LTA_mode in a short period of time in the theoretical corner-case where:

1. The local router spreads information into multiple RA packets, and one of such packets gets lost, thus triggering the LTA mode.

2. The host sends a unicasted solicitation to the local router as part of the LTA mode.

3. The router spreads the response into multiple packets, and e.g. the first of such packets completes all the missing information, thus exiting the LTA mode.

4. One of the remaining RAs of this "batch" would otherwise trigger the LTA mode again.

Thus, the above check only allows the LTA mode to be triggered once every LTA_CYCLE seconds.

Time-driven events:

```
IF LTA_MODE==TRUE:
    TIME=time()

    IF TIME >  (LTA_LAST + LTA_CYCLE)
        Disaasociate any options for which INFO_LAST < LTA_LAST
        LTA_MODE= FALSE
        RS_COUNT= 0

    ELSE IF TIME > (LTA_LAST + RA_WIN + RS_RNDTIME) && TIME >
            (RS_LAST + RS_TIMEOUT) && RS_COUNT < RS_COUNT_MAX:

        IF for all options INFO_LAST >= LTA_LAST
            LTA_MODE= FALSE
            RS_COUNT= 0
        ELSE
            SendRS()
            RS_LAST=TIME
            RS_COUNT++
```

NOTES:

*time() is a monotonically-increasing counter that is incremented once per second, and is employed in this algorithm to measure time.

*SendRS() is a function sends a unicasted Router Solicitation message to the target router (subject to sending rules in [RFC4861]).

RATIONALE:

After a whole LTA_CYCLE has elapsed (i.e., "TIME > (LTA_LAST + LTA_CYCLE)"), SLAAC information that has not been refreshed since the LTA mode was entered should be disassociated with the router for which the LTA algorithm has been performed.

While in the LTA mode, before probing the local router with a unicasted RS, we double-check if all the missing information has been completed/refreshed since the LTA mode was entered. In such case, the LTA mode is exited and the algorithm finished, thus avoiding sending unnecessary RS packets to the local router. Otherwise, a unicasted RS is sent to the local router for which the LTA algorithm is being performed.

[IETF-6MAN-114] illustrates the most common scenarios.

5. IANA Considerations

This document has no actions for IANA.

6. Implementation Status

[NOTE: This section is to be removed by the RFC-Editor before this document is published as an RFC.]

This section summarizes the implementation status of the updates proposed in this document. In some cases, they correspond to variants of the mitigations proposed in this document (e.g., use of reduced default lifetimes for PIOs, albeit using different values than those recommended in this document). In such cases, we believe these implementations signal the intent to deal with the problems described in [RFC8978] while lacking any guidance on the best possible approach to do it.

6.1. More Appropriate Lifetime Values

6.1.1. Router Configuration Variables

6.1.1.1. rad(8)

We have produced a patch for OpenBSD's rad(8) [rad] that employs the default lifetimes recommended in this document, albeit it has not yet been committed to the tree. The patch is available at: <https://www.gont.com.ar/code/fgont-patch-rad-pio-lifetimes.txt>.

6.1.1.2. radvd(8)

The radvd(8) daemon [radvd], normally employed by Linux-based router implementations, currently employs different default lifetimes than those recommended in [RFC4861]. radvd(8) employs the following default values [radvd.conf]:

- *Preferred Lifetime: 14400 seconds (4 hours)
- *Valid Lifetime: 86400 seconds (1 day)

This is not following the specific recommendation in this document, but is already a deviation from the current standards.

6.2. Honor Small PIO Valid Lifetimes

6.2.1. Linux Kernel

A Linux kernel implementation of this document has been committed to the net-next tree. The implementation was produced in April 2020 by Fernando Gont <fgont@si6networks.com>. The corresponding patch can be found at: <https://patchwork.ozlabs.org/project/netdev/patch/20200419122457.GA971@archlinux-current.localdomain/>

6.2.2. NetworkManager

NetworkManager [NetworkManager] processes RA messages with a Valid Lifetime smaller than two hours as recommended in this document.

6.3. Conveying Information in Router Advertisement (RA) Messages

We know of no implementation that splits network configuration information into multiple RA messages.

6.4. Recovery from Stale Configuration Information without Explicit Signaling

6.4.1. dhcpcd(8)

The dhcpcd(8) daemon [dhcpcd], a user-space SLAAC implementation employed by some Linux-based and BSD-derived operating systems, will set the Preferred Lifetime of addresses corresponding to a given prefix to 0 when a single RA from the router that previously advertised the prefix fails to advertise the corresponding prefix. However, it does not affect the corresponding Valid Lifetime. Therefore, it can be considered a partial implementation of this feature.

6.5. Other mitigations implemented in products

[FRITZ] is a Customer Edge Router that tries to deprecate stale prefixes by advertising stale prefixes with a Preferred Lifetime of 0, and a Valid Lifetime of 2 hours (or less). There are two things to note with respect to this implementation:

* Rather than recording prefixes on stable storage (as recommended in [RFC9096]), this implementation checks the source address of IPv6 packets, and assumes that usage of any address that does not correspond to a prefix currently-advertised by the Customer Edge Router is the result of stale network configuration information. Hence, upon receipt of a packet that employs a source address that does not correspond to a currently-advertised prefix, this implementation will start advertising the corresponding prefix with small lifetimes, with the intent of deprecating it.

* Possibly as a result of item "e)" (pp. 19-20) from Section 5.5.3 of [RFC4862] (discussed in Section 4.2 of this document), upon first occurrence of a stale prefix, this implementation will employ a decreasing Valid Lifetime, starting from 2 hours (7200 seconds), as opposed to a Valid Lifetime of 0.

7. Security Considerations

The protocol update in Section 4.2 could allow an on-link attacker to perform a Denial of Service attack against local hosts, by sending a forged RA with a PIO with a Valid Lifetime of 0. Upon receipt of that packet, local hosts would invalidate the corresponding prefix, and therefore remove any addresses configured for that prefix, possibly terminating e.g. associated TCP connections. However, an attacker may achieve similar effects via a number other Neighbor Discovery (ND) attack vectors, such as directing traffic to a non-existing node until ongoing TCP connections time out, or performing a ND-based man-in-the-middle (MITM) attack and subsequently forging TCP RST segments to cause on-going TCP connections to be reset. Thus, for all practical purposes, this attack vector does not really represent any greater risk than other ND attack vectors. As noted in Section 4.2 , in scenarios where RA-based attacks are of concern, proper mitigations such as RA-Guard [RFC6105] [RFC7113] or SEND [RFC3971] should be implemented.

8. Acknowledgments

The authors would like to thank (in alphabetical order) Mikael Abrahamsson, Tore Anderson, Luis Balbinot, Brian Carpenter, Lorenzo Colitti, Owen DeLong, Gert Doering, Thomas Haller, Nick Hilliard, Bob Hinden, Philip Homburg, Lee Howard, Christian Huitema, Tatuya Jinmei, Erik Kline, Ted Lemon, Jen Linkova, Albert Manfredi, Roy Marples, Florian Obser, Jordi Palet Martinez, Michael Richardson, Hiroki Sato, Mark Smith, Hannes Frederic Sowa, Dave Thaler, Tarko Tikan, Ole Troan, Eduard Vasilenko, and Loganaden Velvindron, for providing valuable comments on earlier versions of this document.

The algorithm specified in Section 4.5 is the result of mailing-list discussions over previous versions of this document with Philip Homburg.

Fernando would like to thank Alejandro D'Egidio and Sander Steffann for a discussion of these issues, which led to the publication of [RFC8978], and eventually to this document.

Fernando would also like to thank Brian Carpenter who, over the years, has answered many questions and provided valuable comments that has benefited his protocol-related work.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/

RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[RFC7772] Yourtchenko, A. and L. Colitti, "Reducing Energy Consumption of Router Advertisements", BCP 202, RFC 7772, DOI 10.17487/RFC7772, February 2016, <https://www.rfc-editor.org/info/rfc7772>.

[RFC8028] Baker, F. and B. Carpenter, "First-Hop Router Selection by Hosts in a Multi-Prefix Network", RFC 8028, DOI 10.17487/RFC8028, November 2016, <https://www.rfc-editor.org/info/rfc8028>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8504] Chown, T., Loughney, J., and T. Winters, "IPv6 Node Requirements", BCP 220, RFC 8504, DOI 10.17487/RFC8504, January 2019, <https://www.rfc-editor.org/info/rfc8504>.

9.2. Informative References

[dhcpcd] Marples, R., "dhcpcd - a DHCP client", <https://roy.marples.name/projects/dhcpcd/>.

[FRITZ] Gont, F., "Quiz: Weird IPv6 Traffic on the Local Network (updated with solution)", SI6 Networks Blog, February 2016, <https://www.si6networks.com/2016/02/16/quiz-weird-ipv6-traffic-on-the-local-network-updated-with-solution/>.

[IETF-6MAN-114] Gont, F., Zorz, J., and R. Patterson, "Improving the Robustness of Stateless Address Autoconfiguration (SLAAC) to Flash Renumbering Events", 6man WG meeting IETF 114, 2022, <https://datatracker.ietf.org/meeting/114/materials/slides-114-6man-improving-the-robustness-of-stateless-address-autoconfiguration-slaac-to-flash-renumbering-events-00>.

[IPNG-minutes]
IETF, "IPNG working group (ipngwg) Meeting Minutes", Proceedings of the thirty-eightt Internet Engineering Task Force , April 1997, <https://www.ietf.org/proceedings/38/97apr-final/xrtftr47.htm>.

[NetworkManager] NetworkManager, "NetworkManager web site", <https://wiki.gnome.org/Projects/NetworkManager>.

[rad] Obser, F., "OpenBSD Router Advertisement Daemon - rad(8)", <https://cvsweb.openbsd.org/src/usr.sbin/rad/>.

[radvd] Hawkins, R. and R. Johnson, "Linux IPv6 Router Advertisement Daemon (radvd)", <http://www.litech.org/radvd/>.

[radvd.conf] Hawkins, R. and R. Johnson, "radvd.conf - configuration file of the router advertisement daemon", <https://github.com/reubenhwk/radvd/blob/master/radvd.conf.5.man>.

[RFC1971] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 1971, DOI 10.17487/RFC1971, August 1996, <https://www.rfc-editor.org/info/rfc1971>.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <https://www.rfc-editor.org/info/rfc2827>.

[RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <https://www.rfc-editor.org/info/rfc3971>.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <https://www.rfc-editor.org/info/rfc4191>.

[RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, DOI 10.17487/RFC5927, July 2010, <https://www.rfc-editor.org/info/rfc5927>.

[RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, DOI 10.17487/RFC6105, February 2011, <https://www.rfc-editor.org/info/rfc6105>.

[RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version

6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <https://www.rfc-editor.org/info/rfc6724>.

[RFC7113] Gont, F., "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)", RFC 7113, DOI 10.17487/RFC7113, February 2014, <https://www.rfc-editor.org/info/rfc7113>.

[RFC8106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 8106, DOI 10.17487/RFC8106, March 2017, <https://www.rfc-editor.org/info/rfc8106>.

[RFC8978] Gont, F., Žorž, J., and R. Patterson, "Reaction of IPv6 Stateless Address Autoconfiguration (SLAAC) to Flash-Renumbering Events", RFC 8978, DOI 10.17487/RFC8978, March 2021, <https://www.rfc-editor.org/info/rfc8978>.

[RFC9096] Gont, F., Žorž, J., Patterson, R., and B. Volz, "Improving the Reaction of Customer Edge Routers to IPv6 Renumbering Events", BCP 234, RFC 9096, DOI 10.17487/RFC9096, August 2021, <https://www.rfc-editor.org/info/rfc9096>.

[slaacd] Obser, F., "OpenBSD SLAAC Daemon - slaacd(8)", <https://cvsweb.openbsd.org/src/usr.sbin/slaacd/>.

[systemd] systemd, "systemd web site", <https://systemd.io/>.

Appendix A. Analysis of Some Suggested Workarounds

[This section is to be removed before publication of this document as an RFC].

During the discussion of this document, some alternative workarounds were suggested on the 6man mailing-list. The following subsections analyze these suggested workarounds, in the hopes of avoiding rehashing the same discussions.

A.1. On a Possible Reaction to ICMPv6 Error Messages

It has been suggested that if configured addresses become stale, a CPE enforcing ingress/egress filtering (BCP38) ([RFC2827]) could send ICMPv6 Type 1 (Destination Unreachable) Code 5 (Source address failed ingress/egress policy) error messages to the sending node, and that, upon receipt of such error messages, the sending node could perform heuristics that might help to mitigate the problem discussed in this document.

The aforementioned proposal has a number of drawbacks and limitations:

* It assumes that the CPE routers enforce ingress/egress filtering [RFC2827]. While this is desirable behaviour, it cannot be relied upon.

* It assumes that if the CPE enforces ingress/egress filtering, the CPE will signal the packet drops to the sending node with ICMPv6 Type 1 (Destination Unreachable) Code 5 (Source address failed ingress/egress policy) error messages. While this may be desirable, [RFC2827] does not suggest signaling the packet drops with ICMPv6 error messages, let alone the use of specific error messages (such as Type 1 Code 5) as suggested.

* ICMPv6 Type 1 Code 5 could be interpreted as the employed address being stale, but also as a selected route being inappropriate/ suboptimal. If the later, deprecating addresses or invalidating addresses upon receipt of these error messages would be inappropriate.

* Reacting to these error messages would create a new attack vector that could be exploited from remote networks. This is of particular concern since ICMP-based attacks do not even require that the Source Address of the attack packets be spoofed [RFC5927].

A.2. On a Possible Improvement to Source Address Selection

[RFC6724] specifies source address selection (SAS) for IPv6. Conceptually, it sorts the candidate set of source addresses for a given destination, based on a number of pair-wise comparison rules that must be successively applied until there is a "winning" address.

An implementation might improve source address selection, and prefer the most-recently advertised information. In order to incorporate the "freshness" of information in source address selection, an implementation would be updated as follows:

* The node is assumed to maintain a timer/counter that is updated at least once per second. For example, the time(2) function from unix-like systems could be employed for this purpose.

* The local information associated with each prefix advertised via RAs on the local network is augmented with a "LastAdvertised" timestamp value. Whenever an RA with a PIO with the "A" bit set for such prefix is received, the "LastAdvertised" timestamp is updated with the current value of the timer/counter.

*[RFC6724] is updated such that this rule is incorporated:

Rule 7.5: Prefer fresh information If one of the two source addresses corresponds to a prefix that has been more recently advertised, say LastAdvertised(SA) > LastAdvertised(SA), then prefer that address (SA in our case).

A clear benefit of this approach is that a host will normally prefer "fresh" addresses over possibly stale addresses.

However, there are a number of drawbacks associated with this approach:

*In scenarios where multiple prefixes are being advertised on the same LAN segment, the new SAS rule is *guaranteed* to result in non-deterministic behaviour, with hosts frequently changing the default source address. This is certainly not desirable from a troubleshooting perspective.

*Since the rule must be incorporated before "Rule 8: Use longest matching prefix" from [RFC6724], it may lead to suboptimal paths.

*This new rule may help to improve the selection of a source address, but it does not help with the housekeeping (garbage collection) of configured information:

-If the stale prefix is re-used in another network, nodes employing stale addresses and routes for this prefix will be unable to communicate with the new "owner" of the prefix, since the stale prefix will most likely be considered "on-link".

-Given that the currently recommended default value for the "Valid Lifetime" of PIOs is 2592000 seconds (30 days), it would take too long for hosts to remove the configured addresses and routes for the stale prefix. While the proposed update in Section 4.1 of this document would mitigate this problem, the lifetimes advertised by the local SLAAC router are not under the control of hosts.

As a result, updating IPv6 source address selection does not relieve nodes from improving their SLAAC implementations as specified in Section 4, if at all desirable. On the other hand, the algorithm specified in Section 4.5 would result in Rule 3 of [RFC6724] employing fresh addresses, without leading to non-deterministic behaviour.

Authors' Addresses

Fernando Gont

SI6 Networks
Segurola y Habana 4310, 7mo Piso
Villa Devoto
Ciudad Autonoma de Buenos Aires
Argentina

Email: fgont@si6networks.com
URI: https://www.si6networks.com

Jan Zorz
6connect

Email: jan@connect.com

Richard Patterson
Sky UK

Email: richard.patterson@sky.uk