Internet Engineering Task Force (IETF)
Request for Comments: 7527
Updates: 4429, 4861, 4862
Category: Standards Track
ISSN: 2070-1721

R. Asati
H. Singh
W. Beebee
C. Pignataro
Cisco Systems, Inc.
E. Dart
Lawrence Berkeley National Laboratory
W. George
Time Warner Cable

April 2015

Enhanced Duplicate Address Detection

Abstract

IPv6 Loopback Suppression and Duplicate Address Detection (DAD) are discussed in Appendix A of RFC 4862. That specification mentions a hardware-assisted mechanism to detect looped back DAD messages. If hardware cannot suppress looped back DAD messages, a software solution is required. Several service provider communities have expressed a need for automated detection of looped back Neighbor Discovery (ND) messages used by DAD. This document includes mitigation techniques and outlines the Enhanced DAD algorithm to automate the detection of looped back IPv6 ND messages used by DAD. For network loopback tests, the Enhanced DAD algorithm allows IPv6 to self-heal after a loopback is placed and removed. Further, for certain access networks, this document automates resolving a specific duplicate address conflict. This document updates RFCs 4429, 4861, and 4862.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7527.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

IPv6 Loopback Suppression and Duplicate Address Detection (DAD) are discussed in Appendix A of [RFC4862]. That specification mentions a hardware-assisted mechanism to detect looped back DAD messages. If hardware cannot suppress looped back DAD messages, a software solution is required. One specific DAD message is the Neighbor Solicitation (NS), specified in [RFC4861]. The NS is issued by the network interface of an IPv6 node for DAD. Another message involved in DAD is the Neighbor Advertisement (NA). The Enhanced DAD algorithm specified in this document focuses on detecting an NS looped back to the transmitting interface during the DAD operation. Detecting a looped back NA does not solve the looped back DAD

problem. Detection of any other looped back ND messages during the DAD operation is outside the scope of this document. This document also includes a section on mitigation that discusses means already available to mitigate the DAD loopback problem. This document updates RFCs 4429, 4861, and 4862. It updates RFCs 4429 and 4862 to use the Enhanced DAD algorithm to detect looped back DAD probes, and it updates RFC 4861 as described in Section 4.3 below.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.2. Terminology

- DAD-failed state - Duplication Address Detection failure as specified in [RFC4862]. Note even Optimistic DAD as specified in [RFC4429] can fail due to a looped back DAD probe. This document covers looped back detection for Optimistic DAD as well.

- Looped back message - also referred to as a reflected message. The message sent by the sender is received by the sender due to the network or an upper-layer protocol on the sender looping the message back.

- Loopback - A function in which the router’s Layer 3 interface (or the circuit to which the router’s interface is connected) is looped back or connected to itself. Loopback causes packets sent by the interface to be received by the interface and results in interface unavailability for regular data traffic forwarding. See more details in Section 9.1 of [RFC2328]. The Loopback function is commonly used in an interface context to gain information on the quality of the interface, by employing mechanisms such as ICMPv6 pings and bit-error tests. In a circuit context, this function is used in wide-area environments including optical Dense Wavelength Division Multiplexing (DWDM) and Synchronous Optical Network / Synchronous Digital Hierarchy (SONET/SDH) for fault isolation (e.g., by placing a loopback at different geographic locations along the path of a wide-area circuit to help locate a circuit fault). The Loopback function may be employed locally or remotely.

- NS(DAD) - shorthand notation to denote a Neighbor Solicitation (NS) (as specified in [RFC4861]) that has an unspecified IPv6 source address and was issued during DAD.

2. Problem Statement

Service providers have reported a problem with DAD that arises in a few scenarios. In the first scenario, loopback testing for troubleshooting purposes is underway on a circuit connected to an IPv6-enabled interface on a router. The interface issues an NS for the IPv6 link-local address DAD. The NS is reflected back to the router interface due to the loopback condition of the circuit, and the router interface enters a DAD-failed state. After the loopback condition is removed, IPv4 will return to operation without further manual intervention. However, IPv6 will remain in DAD-failed state until manual intervention on the router restores IPv6 to operation.

In the second scenario, two broadband modems are served by the same service provider and terminate to the same Layer 3 interface on an IPv6-enabled access concentrator. In this case, the two modems' Ethernet interfaces are also connected to a common local network (collision domain). The access concentrator serving the modems is the first-hop IPv6 router for the modems and issues a NS(DAD) message for the IPv6 link-local address of its Layer 3 interface. The NS message reaches one modem first, and this modem sends the message to the local network, where the second modem receives the message and then forwards it back to the access concentrator. The looped back NS message causes the network interface on the access concentrator to be in a DAD-failed state. Such a network interface typically serves thousands of broadband modems, and all would have their IPv6 connectivity affected until the DAD-failed state is cleared. Additionally, it may be difficult for the user of the access concentrator to determine the source of the looped back DAD message. Thus, in order to avoid IPv6 outages that can potentially affect multiple users, there is a need for automated detection of looped back NS messages during DAD operations by a node.

Note: In both examples above, the IPv6 link-local address DAD operation fails due to a looped back DAD probe. However, the problem of a looped back DAD probe exists for any IPv6 address type including global addresses.

3. Operational Mitigation Options

Two mitigation options are described below that do not require any change to existing implementations.

3.1. Disable DAD on an Interface

One can disable DAD on an interface so that there are no NS(DAD) messages issued. While this mitigation may be the simplest, the mitigation has three drawbacks: 1) care is needed when making such

configuration changes on point-to-point interfaces, 2) this is a one-time manual configuration on each interface, and 3) genuine duplicates on the link will not be detected.

A service provider router, such as an access concentrator, or network core router, SHOULD support the DAD deactivation per interface.

3.2. Dynamic Disable/Enable of DAD Using Layer 2 Protocol

Some Layer 2 protocols include provisions to detect the existence of a loopback on an interface circuit, usually by comparing protocol data sent and received. For example, the Point-to-Point Protocol (PPP) uses a magic number (Section 6.4 of [RFC1661]) to detect a loopback on an interface.

When a Layer 2 protocol detects that a loopback is present on an interface circuit, the device MUST temporarily disable DAD on the interface. When the protocol detects that a loopback is no longer present (or the interface state has changed), the device MUST (re-)enable DAD on that interface.

This mitigation has several benefits. It leverages the Layer 2 protocol's built-in hardware loopback detection capability, if available. Being a hardware solution, it scales better than the software solution proposed in this document. This mitigation also scales better since it relies on an event-driven model that requires no additional state or timer. This may be significant on devices with hundreds or thousands of interfaces that may be in loopback for long periods of time (e.g., awaiting turn-up).

Detecting looped back DAD messages using a Layer 2 protocol SHOULD be enabled by default, and it MUST be a configurable option if the Layer 2 technology provides means for detecting loopback messages on an interface circuit.

3.3. Operational Considerations

The mitigation options discussed above do not require the devices on both ends of the circuit to support the mitigation functionality simultaneously and do not propose any capability negotiation. They are effective for unidirectional circuit or interface loopback (i.e. the loopback is placed in one direction on the circuit, rendering the other direction nonoperational), but they may not be effective for a bidirectional loopback (i.e., the loopback is placed in both directions of the circuit interface, so as to identify the faulty segment). This is because, unless both ends followed a mitigation

option specified in this document, the noncompliant device would follow current behavior and disable IPv6 on that interface due to DAD until manual intervention restores it.

4. The Enhanced DAD Algorithm

The Enhanced DAD algorithm covers detection of a looped back NS(DAD) message. This document proposes use of a random number in the Nonce Option specified in SEcure Neighbor Discovery (SEND) [RFC3971]. Note [RFC3971] does not provide a recommendation for pseudorandom functions. Pseudorandom functions are covered in [RFC4086]. Since a nonce is used only once, the NS(DAD) for each IPv6 address of an interface uses a different nonce. Additional details of the algorithm are included in Section 4.1.

If there is a collision because two nodes used the same Target Address in their NS(DAD) and generated the same random nonce, then the algorithm will incorrectly detect a looped back NS(DAD) when a genuine address collision has occurred. Since each looped back NS(DAD) event is logged to system management, the administrator of the network will have access to the information necessary to intervene manually. Also, because the nodes will have detected what appear to be looped back NS(DAD) messages, they will continue to probe, and it is unlikely that they will choose the same nonce the second time (assuming quality random number generators).

The algorithm is capable of detecting any ND solicitation (NS and Router Solicitation) or advertisement (NA and Router Advertisement) that is looped back. However, there may be increased implementation complexity and memory usage for the sender node to store a nonce and nonce-related state for all ND messages. Therefore, this document does not recommend using the algorithm outside of the DAD operation by an interface on a node.

4.1. Processing Rules for Senders

If a node has been configured to use the Enhanced DAD algorithm, when sending an NS(DAD) for a tentative or optimistic interface address, the sender MUST generate a random nonce associated with the interface address, MUST store the nonce internally, and MUST include the nonce in the Nonce option included in the NS(DAD). If the interface does not receive any DAD failure indications within RetransTimer milliseconds (see [RFC4861]) after having sent DupAddrDetectTransmits Neighbor Solicitations, the interface moves the Target Address to the assigned state.

If any probe is looped back within RetransTimer milliseconds after having sent DupAddrDetectTransmits NS(DAD) messages, the interface continues with another MAX_MULTICAST_SOLICIT number of NS(DAD) messages transmitted RetransTimer milliseconds apart. Section 2 of [RFC3971] defines a single-use nonce, so each Enhanced DAD probe uses a different nonce. If no probe is looped back within RetransTimer milliseconds after MAX_MULTICAST_SOLICIT NS(DAD) messages are sent, the probing stops. The probing MAY be stopped via manual intervention. When probing is stopped, the interface moves the Target Address to the assigned state.

4.2. Processing Rules for Receivers

If the node has been configured to use the Enhanced DAD algorithm and an interface on the node receives any NS(DAD) message where the Target Address matches the interface address (in tentative or optimistic state), the receiver compares the nonce included in the message, with any stored nonce on the receiving interface. If a match is found, the node SHOULD log a system management message, SHOULD update any statistics counter, and MUST drop the received message. If the received NS(DAD) message includes a nonce and no match is found with any stored nonce, the node SHOULD log a system management message for a DAD-failed state and SHOULD update any statistics counter.

4.3. Changes to RFC 4861

The following text is appended to the Source Address definition in Section 4.3 of [RFC4861]:

If a node has been configured to use the Enhanced DAD algorithm, an NS with an unspecified source address adds the Nonce option to the message and implements the state machine of the Enhanced DAD algorithm.

The following text is appended to the RetransTimer variable description in Section 6.3.2 of [RFC4861]:

The RetransTimer MAY be overridden by a link-specific document if a node supports the Enhanced DAD algorithm.

5. Action to Perform on Detecting a Genuine Duplicate

As described in the paragraphs above, the nonce can also serve to detect genuine duplicates even when the network has potential for looping back ND messages. When a genuine duplicate is detected, the node follows the manual intervention specified in Section 5.4.5 of [RFC4862]. However, in certain cases, if the genuine duplicate

matches the tentative or optimistic IPv6 address of a network interface of the access concentrator, additional automated action is recommended.

Some networks follow a trust model where a trusted router serves untrusted IPv6 host nodes. Operators of such networks have a desire to take automated action if a network interface of the trusted router has a tentative or optimistic address duplicated by a host. One example of a type of access network is cable broadband deployment where the access concentrator is the first-hop IPv6 router to multiple broadband modems and supports proxying of DAD messages. The network interface on the access concentrator initiates DAD for an IPv6 address and detects a genuine duplicate due to receiving an NS(DAD) or an NA message. On detecting such a duplicate, the access concentrator SHOULD log a system management message, drop the received ND message, and block the modem on whose Layer 2 service identifier the duplicate NS(DAD) or NA message was received. Any other network that follows the same trust model MAY use the automated action proposed in this section.

6. Security Considerations

This document does not improve or reduce the security posture of [RFC4862]. The nonce can be exploited by a rogue deliberately changing the nonce to fail the looped back detection specified by the Enhanced DAD algorithm. SEND is recommended to circumvent this exploit. Additionally, the nonce does not protect against the DoS caused by a rogue node replying by a fake NA to all DAD probes. SEND is recommended to circumvent this exploit also. Disabling DAD has an obvious security issue before a remote node on the link can issue reflected NS(DAD) messages. Again, SEND is recommended for this exploit. Source Address Validation Improvement (SAVI) [RFC6620] also protects against various attacks by on-link rogues.

7. Normative References

[RFC1661] Simpson, W., Ed., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994, <http://www.rfc-editor.org/info/rfc1661>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998, <http://www.rfc-editor.org/info/rfc2328>.

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

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005, <http://www.rfc-editor.org/info/rfc4086>.

[RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, April 2006, <http://www.rfc-editor.org/info/rfc4429>.

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

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

[RFC6620] Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses", RFC 6620, May 2012, <http://www.rfc-editor.org/info/rfc6620>.

Acknowledgements

Thanks (in alphabetical order by first name) to Adrian Farrel, Benoit Claise, Bernie Volz, Brian Haberman, Dmitry Anipko, Eric Levy-Abegnoli, Eric Vyncke, Erik Nordmark, Fred Templin, Hilarie Orman, Jouni Korhonen, Michael Sinatra, Ole Troan, Pascal Thubert, Ray Hunter, Suresh Krishnan, Tassos Chatzithomaoglou, and Tim Chown for their guidance and review of the document. Thanks to Thomas Narten for encouraging this work. Thanks to Steinar Haug and Scott Beuker for describing some of the use cases.

Authors' Addresses

Rajiv Asati
Cisco Systems, Inc.
7025 Kit Creek road
Research Triangle Park, NC 27709-4987
United States

EMail: rajiva@cisco.com
URI: http://www.cisco.com/

Hemant Singh
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, MA 01719
United States

Phone: +1 978 936 1622
EMail: shemant@cisco.com
URI: http://www.cisco.com/

Wes Beebee
Cisco Systems, Inc.
1414 Massachusetts Ave.
Boxborough, MA 01719
United States

Phone: +1 978 936 2030
EMail: wbeebee@cisco.com
URI: http://www.cisco.com/

Carlos Pignataro
Cisco Systems, Inc.
7200-12 Kit Creek Road
Research Triangle Park, NC 27709
United States

EMail: cpignata@cisco.com
URI: http://www.cisco.com/

Eli Dart
Lawrence Berkeley National Laboratory
1 Cyclotron Road, Berkeley, CA 94720
United States

EMail: dart@es.net
URI: http://www.es.net/

Wesley George
Time Warner Cable
13820 Sunrise Valley Drive
Herndon, VA 20171
United States

EMail: wesley.george@twcable.com