Workgroup: Dynamic Host Configuration
Internet-Draft:
draft-wkumari-dhc-addr-notification-00
Published: 7 March 2022
Intended Status: Experimental
Expires: 8 September 2022
Authors: W. Kumari
Google, LLC
S. Krishnan
Kaloom
S. Jiang
R. Asati
Cisco Systems, Inc.

Registering Self-generated IPv6 Addresses using DHCPv6

Abstract

This document defines a method to inform a DHCPv6 server that a device has a self-generated or statically configured address.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://wkumari.github.io/draft-wkumari-dhc-addr-notification/draft-wkumari-dhc-addr-notification.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-wkumari-dhc-addr-notification/.

Discussion of this document takes place on the Dynamic Host Configuration Working Group mailing list (mailto:dhcwg@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/dhcwg/.

Source for this draft and an issue tracker can be found at https://github.com/wkumari/draft-wkumari-dhc-addr-notification.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 September 2022.

Table of Contents

1. Introduction

It is very common operational practice, especially in enterprise networks, to use IPv4 DHCP logs for troubleshooting or security purposes. Examples of this include a helpdesk dealing with a ticket such as "The CEO's laptop cannot connect to the printer"; if the MAC address of the printer is known (for example from an inventory system), the IPv4 address can be retrieved from the DHCP logs and the printer pinged to determine if it is reachable. Another common example is a Security Operations team discovering suspicious events in outbound firewall logs and then consulting DHCP logs to determine

which employee's laptop had that IPv4 address at that time so that they can quarantine it and remove the malware.

This operational practice relies on the DHCP server knowing the IP address assignments. Therefore, the practice does not work if static IP addresses are manually configured on devices or self-assigned addresses (such as when self-configuring an IPv6 address using SLAAC [RFC4862]) are used.

The lack of this parity with IPv4 is one of the reasons that some enterprise networks are unwilling to deploy IPv6.

This document provides a mechanism for a device to inform the DHCPv6 server that it has a self-configured IPv6 address (or has a statically configured address), and thus provides parity with IPv4 in this aspect.

This document borrows heavily from a previous document, draft-ietf-dhc-addr-registration, which defined "a mechanism to register self-generated and statically configured addresses in DNS through a DHCPv6 server".

2. Conventions and Definitions

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. Description of Mechanism

After successfully assigning a self-generated IPv6 address on one of its interfaces, an end-host implementing this specification **SHOULD** send an ADDR-REG-NOTIFICATION message to a DHCPv6 address registration server. After receiving the address registration request, the DHCPv6 server records and logs the IPv6 address. An acknowledgement **MUST** be sent back to the end host to indicate whether or not the registration operation succeeded.

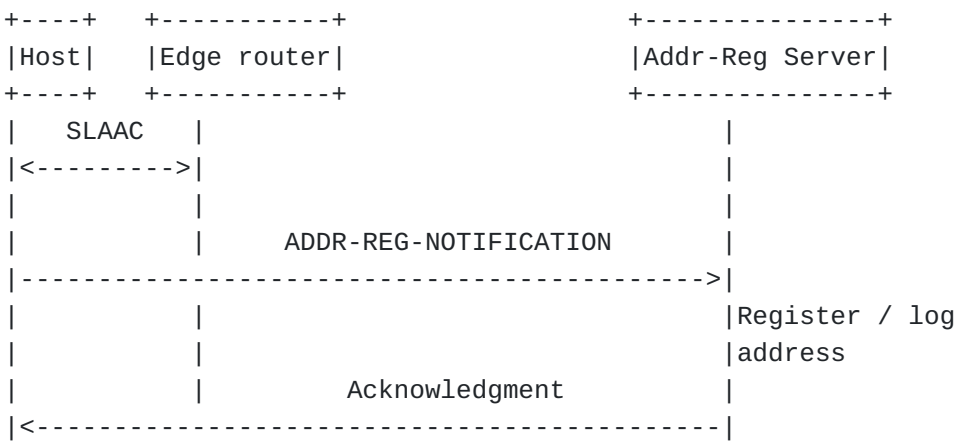
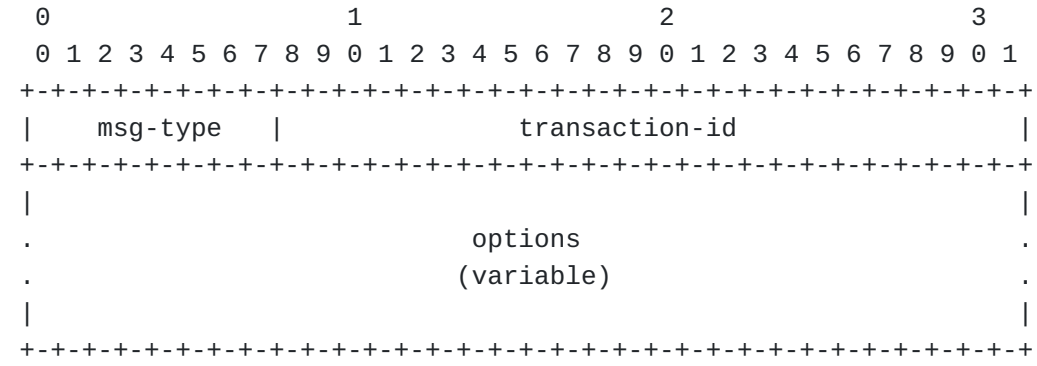


Figure 1: Address Registration ProcedureAddress Registration Procedure

The registration server **MAY** apply certain filter/accept criteria for address registration requests (for example to deny registration of addresses that are not appropriate for the link, etc.)

4. DHCPv6 ADDR-REG-NOTIFICATION Message

The DHCPv6 client sends an ADDR-REG-NOTIFICATION message to a server to request that the use of this address be registered and logged. The format of the ADDR-REG-NOTIFICATION message is described as follows:



msg-type — Identifies the DHCPv6 message type; Set to ADDR-REG-NOTIFICATION (TBA1).

transaction-id — The transaction ID for this message exchange.

options — Options carried in this message.

Figure 2: DHCPv6 ADDR-REG-NOTIFICATION message

The ADDR-REG-NOTIFICATION message **MUST NOT** contain server-identifier option and **MUST** contain the IA Address option. The ADDR-REG-NOTIFICATION message is dedicated for clients to initiate an address

registration request toward an address registration server. Consequently, clients **MUST NOT** put any Option Request Option(s) in the ADDR-REG-NOTIFICATION message.

Clients **MUST** discard any received ADDR-REG-NOTIFICATION messages.

Servers **MUST** discard any ADDR-REG-NOTIFICATION messages that meet any of the following conditions:

- *the message does not include a Client Identifier option;
- *the message includes a Server Identifier option;
- *the message does not include at least one IA Address option;
- *the message includes an Option Request Option.

5. DHCPv6 Address Registration Procedure

The DHCPv6 protocol is used as the address registration protocol when a DHCPv6 server performs the role of an address registration server. The DHCPv6 IA Address option [RFC3315]is adopted in order to fulfill the address registration interactions.

5.1. DHCPv6 Address Registration Request

The end-host sends a DHCPv6 ADDR-REG-NOTIFICATION message to the address registration server to the All_DHCP_Relay_Agents_and_Servers multicast address (ff02::1:2). The host **SHOULD** send the packet from the address being registered.

The end-host **MUST** include a Client Identifier option and at least one IA Address option in the ADDR-REG-NOTIFICATION message. The host **SHOULD** send separate messages for each address (so each message include only one IA Address option) but **MAY** send a single packet containing multiple options.

The host **MUST NOT** send the ADDR-REG-NOTIFICATION message for addresses which are not in "preferred" (RFC4862) state.

{TODO (WK): DHCPv6 uses "DHCP Unique Identifier (DUID)" to identify clients. This doesn't really meet our design goal of "what IP does the printer have?!". One of the DUID types is "DUID Based on Link-layer Address (DUID-LL)", but this is "any one network interface(s)" - this is probably good enough for the inventory use case, but still not ideal}

After receiving this ADDR-REG-NOTIFICATION message, the address registration server **MUST** register the binding between the provided Client Identifier and IPv6 address. If the DHCPv6 server does not

support the address registration function, it **MUST** drop the message (and may log the event).

5.2. Registration Expiry and Refresh

For every successful binding registration, the address registration server **MUST** record the Client-Identifier-to-IPv6-address bindings and associated valid-lifetimes in its storage, and **SHOULD** log this information in a manner similar to if it had performed the assignment.

If a ADDR-REG-NOTIFICATION message updates the existing Client-Identifier-to-IPv6-address binding the server **MAY** log the event.

The address registration client **MUST** refresh the registration before it expires (i.e. before the preferred lifetime of the IA address elapses) by sending a new ADDR-REG-NOTIFICATION to the address registration server. If the address registration server does not receive such a refresh after the preferred lifetime has passed, it **SHOULD** remove the record of the Client-Identifier-to-IPv6-address binding.

It is **RECOMMENDED** that clients initiate a refresh at about 85% of the preferred lifetime. Because RAs may periodically 'reset' the preferred- lifetime, the refresh timer **MUST** be independently maintained from the address valid-lifetime. Clients **SHOULD** set a refresh timer to 85% of the preferred lifetime when they complete a registration operation and only update this timer if 85% of any updated preferred lifetime would be sooner than the timer.

{TODO: is the preferred lifetime a good idea? The default value is 7 days which seems rather long. Indeed we might say that it's an administrator's job to configure non-default lifetime... Also, what about statically assigned addresses or PIOs with the inifinite lifetime??}

5.3. Acknowledging Registration and Retransmission

After an address registration server accepts an address registration request, it **MUST** send a Reply message as the response to the client. The acceptance reply only means that the server has taken responsibility to remember and log the client, not that it has yet done so.

The server generates a Reply message and includes a Status Code option with value Success, a Server Identifier option with the server's DUID, and a Client Identifier option with the client's DUID.

If there is no reply received within some interval, the client **SHOULD** retransmit the message according to section 14 of [RFC3315], using the following parameters:

- *IRT ADDR_REG_TIMEOUT
- *MRT ADDR_REG_MAX_RT
- *MRC ADDR_REG_MAX_RC
- *MRD 0

The below presents a table of values used to describe the message transmission behavior of clients and servers:

Parameter	Default	Description
ADDR_REG_TIMEOUT	1 secs	Initial Addr Registration Request timeout
ADDR_REG_MAX_RT	60 secs	Max Addr Registration Request timeout value
ADDR_REG_MAX_RC	5	Max Request retry attempts

For each IA Address option in the ADDR-REG-NOTIFICATION message for which the server does not accept its associated registration request, the server adds an IA Address option with the associated IPv6 address, and includes a Status Code option with the value RegistrationDenied (TBA2) in the IA Address option. No other options are included in the IA Address option.

Upon receiving a RegistrationDenied error status code, the client **MAY** also resend the message following normal retransmission routines defined in [RFC3315] with above parameters. The client **MUST** wait out the retransmission time before retrying.

6. Security Considerations

An attacker may attempt to register a large number of addresses in quick succession in order to overwhelm the address registration server and / or fill up log files. These attacks may be mitigated by using generic DHCPv6 protection such as the AUTH option [RFC3315].

One of the primary use-cases for the mechanism described in this document is to identify which device is infected with malware (or is otherwise doing bad things) so that it can be blocked from accessing the network. As the device itself is responsible for informing the DHCPv6 server that it is using an address, malware (or a malicious client) can simply not send the ADDR-REG-NOTIFICATION message. This is an informational, optional mechanism, and is designed to aid in debugging. It is not intended to be a strong security access mechanism.

7. IANA Considerations

This document defines a new DHCPv6 message, the ADDR-REG-NOTIFICATION message (TBA1) described in Section 4, that requires an allocation out of the registry of Message Types defined at http://www.iana.org/assignments/dhcpv6-parameters/

7.1. Value Description Reference

TBA1 ADDR-REG-NOTIFICATION this document

This document defines a new DHCPv6 Status code, the RegistrationDenied (TBA2) described in Section 5, that requires an allocation out of the registry of Status Codes defined at http://www.iana.org/assignments/dhcpv6-parameters/

7.2. Code Name Reference

TBA2 RegistrationDenied this document

8. References

8.1. 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/rfc/rfc2119>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <https://www.rfc-editor.org/rfc/rfc3315>.

[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/rfc/rfc8174>.

8.2. Informative References

[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/rfc/rfc4862>.

Acknowledgments

"We've Been Trying To Reach You About Your Car's Extended Warranty"

Much thanks to Jen Linkova for additional text on client behavior.
Also, much thanks to Erik Kline and Lorenzo Colitti for significant
discussion and feedback.

Contributors

Gang Chen
China Mobile
53A, Xibianmennei Ave.
Xuanwu District
Beijing
P.R. China

Email: phdgang@gmail.com

Authors' Addresses

Warren Kumari
Google, LLC

Email: warren@kumari.net

Suresh Krishnan
Kaloom

Email: suresh@kaloom.com

Sheng Jiang
Beijing
P.R. China

Email: jiangsheng@gmail.com

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

Email: rajiva@cisco.com