Dynamic Host Configuration (dhc)
Internet-Draft
Intended status: Standards Track
Expires: April 4, 2015
F. Gont
SI6 Networks / UTN-FRH
W. Liu
Huawei Technologies
October 1, 2014

A Method for Generating Semantically Opaque Interface Identifiers with Dynamic Host Configuration Protocol for IPv6 (DHCPv6)

draft-ietf-dhc-stable-privacy-addresses-00

Abstract

This document specifies a method for selecting IPv6 Interface Identifiers, to be employed by Dynamic Host Configuration Protocol for IPv6 (DHCPv6) servers when leasing non-temporary IPv6 addresses to DHCPv6 clients. This method is a DHCPv6 server side algorithm, that does not require any updates to the existing DHCPv6 specifications. The aforementioned method results in stable addresses within each subnet, even in the presence of multiple DHCPv6 servers or even DHCPv6 server reinstallments. It is a DHCPv6-variant of the method specified in RFC 7217 for IPv6 Stateless Address Autoconfiguration.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 4, 2015.

Copyright Notice

Copyright (c) 2014 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

Stable IPv6 addresses tend to simplify event logging, trouble-shooting, enforcement of access controls and quality of service, etc. However, there are a number of scenarios in which a host employing the DHCPv6 protocol [RFC3315] may be assigned different IPv6 addresses for the same interface within the same subnet over time. For example, this may happen when multiple servers operate on the same network to provide increased availability, but may also happen as a result of DHCPv6 server reinstallments and other scenarios.

This document specifies a method for selecting IPv6 Interface Identifiers, to be employed by Dynamic Host Configuration Protocol for IPv6 (DHCPv6) servers when leasing non-temporary IPv6 addresses to DHCPv6 clients (i.e., to be employed with IA_NA options). This method is a DHCPv6 server side algorithm, that does not require any updates to the existing DHCPv6 specifications. The aforementioned method has the following properties:

- The resulting IPv6 addresses remain stable within each subnet for the same network interface of the same client, even when different DHCPv6 servers (implementing this specification) are employed.

o It must be difficult for an outsider to predict the IPv6 addresses that will be generated by the method specified in this document, even with knowledge of the IPv6 addresses generated for other nodes within the same network.

The method specified in this document achieves the aforementioned goals by means of a calculated technique as opposed to e.g. state-sharing among DHCPv6 servers . This approach has been already suggested in [RFC7031]. We note that the method specified in this document is essentially a DHCPv6-version of the "Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)" specified in [RFC7217].

2. Terminology

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 RFC 2119 [RFC2119].

3. Method Specification

DHCPv6 server implementations conforming to this specification MUST generate non-temporary IPv6 addresses using the algorithm specified in this section.

Implementations conforming to this specification SHOULD provide the means for a system administrator to enable or disable the use of this algorithm for generating IPv6 addresses.

Unless otherwise noted, all of the parameters included in the expression below MUST be included when generating an IPv6 address.

1. Compute a random (but stable) identifier with the expression:

   RID = F(Prefix | Client_DUID | IAID | Counter | secret_key)

   Where:

   RID:
      Random (but stable) Identifier

F():
A pseudorandom function (PRF) that MUST NOT be computable from the outside (without knowledge of the secret key). F() MUST also be difficult to reverse, such that it resists attempts to obtain the secret_key, even when given samples of the output of F() and knowledge or control of the other input parameters. F() SHOULD produce an output of at least 64 bits. F() could be implemented as a cryptographic hash of the concatenation of each of the function parameters. The default algorithm to be employed for F() SHOULD be SHA-1 [FIPS-SHS]. An implementation MAY provide the means for selecting other other

algorithms (e.g., SHA-256) for F(). Note: MD5 [RFC1321] is considered unacceptable for F() [RFC6151].

|:
An operator representing "concatenation".

Prefix:
A prefix that represents an IPv6 address pool from which the DHCPv6 server will assign addresses. That is, this algorithm REQUIRES that the DHCPv6 server manages all the IPv6 address space within a specified prefix (as opposed to, e.g., an address range that cannot be represented with a prefix notation) and that it can be configured with such a prefix. If multiple servers operate on the same network to provide increased availability, all such DHCPv6 servers MUST be configured with the same Prefix. It is the administrator's responsibility that the aforementioned requirement is met.

Client_DUID:
The DUID value contained in the Client Identifier option received in the client message.

IAID:
The IAID value contained in the IA_NA option received in the client message.

Counter:
A variable that is employed to resolve address conflicts. It MUST be initialized to 0.

secret_key:
A secret key configured by the DHCPv6 server administrator, which MUST NOT be known by the attacker. An implementation of this specification MUST provide an interface for viewing and changing the secret key. All DHCPv6 servers leasing addresses from the same Prefix MUST employ the same secret key.

2. The Interface Identifier is obtained by taking as many bits from the RID value (computed in the previous step) as necessary, starting from the least significant bit.

   We note that [RFC4291] requires that, the Interface IDs of all unicast addresses (except those that start with the binary value 000) be 64-bit long. However, the method discussed in this document could be employed for generating Interface IDs of any arbitrary length, albeit at the expense of reduced entropy (when employing Interface IDs smaller than 64 bits).

   The resulting Interface Identifier MUST be compared against the reserved IPv6 Interface Identifiers [RFC5453] [IANA-RESERVED-IID]. In the event that an unacceptable identifier has been generated, the Counter variable should be incremented by 1, and a new Interface ID should be computed with the updated Counter value.

3. The IPv6 address is finally obtained by concatenating the Prefix with the Interface Identifier obtained in the previous step. If the resulting address is not available (e.g., there is a conflicting binding), the server should increment the Counter variable, and a new Interface ID and IPv6 address should be computed with the updated Counter value.

This document requires that SHA-1 be the default function to be used for F(), such that, all other configuration parameters being the same, different implementations of this specification result in the same IPv6 addresses.

Including the Prefix in the PRF computation causes the Interface Identifier to for each address from a different prefix assigned to the same client. This mitigates the correlation of activities of

multi-homed nodes (since each of the corresponding addresses will employ a different Interface ID), host-tracking (since the network prefix will change as the node moves from one network to another), and any other attacks that benefit from predictable Interface Identifiers (such as IPv6 address scanning attacks) [I-D.ietf-6man-ipv6-address-generation-privacy].

As required by [RFC3315], an IAID is associated with each of the client's network interfaces, and is consistent across restarts of the DHCP client.

The Counter parameter provides the means to intentionally cause this algorithm to produce a different IPv6 addresses (all other parameters being the same). This could be necessary to resolve address conflicts (e.g. the resulting address having a conflicting binding).

Note that the result of F() in the algorithm above is no more secure than the secret key. If an attacker is aware of the PRF that is being used by the DHCPv6 server (which we should expect), and the attacker can obtain enough material (i.e. addresses generated by the DHCPv6 server), the attacker may simply search the entire secret-key space to find matches. To protect against this, the secret key SHOULD be of at least 128 bits. Key lengths of at least 128 bits should be adequate.

Providing a mechanism to display and change the secret_key is crucial for having different DHCPv6 servers produce the same IPv6 addresses, and for causing a replacement system to generate the same IPv6 addresses as the system being replaced. We note that since the privacy of the scheme specified in this document relies on the secrecy of the secret_key parameter, implementations should constrain access to the secret_key parameter to the extent practicable (e.g., require superuser privileges to access it). Furthermore, in order to prevent leakages of the secret_key parameter, it should not be used for any other purposes than being a parameter to the scheme specified in this document.

We note that all of the bits in the resulting Interface IDs are treated as "opaque" bits [RFC7136]. For example, the universal/local bit of Modified EUI-64 format identifiers is treated as any other bit

of such identifier.

4. IANA Considerations

There are no IANA registries within this document. The RFC-Editor can remove this section before publication of this document as an RFC.

5. Security Considerations

The method specified in this document results in IPv6 Interface Identifiers (and hence IPv6 addresses) that do not follow any specific pattern. Thus, address-scanning attacks [I-D.ietf-opsec-ipv6-host-scanning] are mitigated.

The method specified in this document neither mitigates nor exacerbates the security considerations for DHCPv6 discussed in [RFC3315].

6. Acknowledgements

This document is based on [RFC7217], authored by Fernando Gont.

The authors would like to thank Tatuya Jinmei for providing valuable comments on earlier versions of this documents.

The authors would like to thank Ted Lemon, who kindly answered some DHCPv6-related questions.

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.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6

(IPv6) Specification", RFC 2460, December 1998.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC5453] Krishnan, S., "Reserved IPv6 Interface Identifiers", RFC 5453, February 2009.

[RFC7136] Carpenter, B. and S. Jiang, "Significance of IPv6 Interface Identifiers", RFC 7136, February 2014.

7.2. Informative References

[FIPS-SHS]
FIPS, , "Secure Hash Standard (SHS)", Federal Information Processing Standards Publication 180-4, March 2012, <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>.

[I-D.ietf-6man-ipv6-address-generation-privacy]
Cooper, A., Gont, F., and D. Thaler, "Privacy Considerations for IPv6 Address Generation Mechanisms", draft-ietf-6man-ipv6-address-generation-privacy-01 (work in progress), February 2014.

[I-D.ietf-opsec-ipv6-host-scanning]
Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", draft-ietf-opsec-ipv6-host-scanning-04 (work in progress), June 2014.

[IANA-RESERVED-IID]
Reserved IPv6 Interface Identifiers, , "http://www.iana.org/assignments/ipv6-interface-ids/ipv6-interface-ids.xml", .

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321,

April 1992.

[RFC6151] Turner, S. and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms", RFC 6151, March 2011.

[RFC7031] Mrugalski, T. and K. Kinnear, "DHCPv6 Failover Requirements", RFC 7031, September 2013.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, April 2014.

Authors' Addresses

Fernando Gont
SI6 Networks / UTN-FRH
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: http://www.si6networks.com

Will(Shucheng) Liu
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Email: liushucheng@huawei.com