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

Abstract

   This document describes a method for selecting IPv6 Interface
   Identifiers that can 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 DHCPv6 server reinstallments. It is a DHCPv6 variant of
   the method specified in RFC 7217 for IPv6 Stateless Address
   Autoconfiguration.

IESG Note

   A predecessor to this document was earlier a working group document
   in the DHC WG. The WG decided to stop further work in this area
   because such work was not considered useful.

   The proposal described in this document has an unaddressed failure
   case that makes it unsuitable for use as the mechanism to provide the
   claimed failover features for DHCPv6 servers. Specifically, when a
   DHCPv6 client DECLINEs a provided address there is no recovery
   mechanism described that will result in the DHCPv6 client obtaining a
   working IPv6 address.

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for informational purposes.

   This is a contribution to the RFC Series, independently of any other
   RFC stream. The RFC Editor has chosen to publish this document at
   its discretion and makes no statement about its value for
   implementation or deployment. Documents approved for publication by
   the RFC Editor are not a candidate for any level of Internet
   Standard; see Section 2 of RFC 7841.

   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/rfc7943.

Table of Contents

1. Introduction

   The benefits of stable IPv6 addresses are discussed in [RFC7721].
   Providing address stability across server reinstallations or when a
   database of previous DHCPv6 address leases is unavailable is of use
   not only when a DHCPv6 server must be reinstalled or the address-
   lease database becomes corrupted, but is also of use when
   implementation constraints (e.g., a DHCPv6 server implementation on
   an embedded device) make it impossible for a DHCPv6 server
   implementation to maintain a database of previous DHCPv6 address
   leases. Additionally, [RFC7031] describes scenarios where multiple
   DHCPv6 servers are required to run in such a way as to provide
   increased availability in case of server failures.

   This document describes a method for selecting IPv6 Interface
   Identifiers that can be employed by 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:

   o 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 Predicting 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,
      becomes very difficult.

   The method specified in this document achieves the aforementioned
   properties by means of a calculated technique as opposed to, e.g.,
   state sharing among DHCPv6 servers. This approach has already been
   suggested in [RFC7031]. We note that the method described 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. Applicability and Design Goals

   This document simply describes one possible approach for selecting
   IPv6 Interface Identifiers to be employed by DHCPv6 servers when
   leasing non-temporary IPv6 addresses to DHCPv6 clients, with the
   following properties:

   o The resulting IPv6 addresses remain stable within each subnet for
      the same network interface of the same client.

   o The resulting IPv6 addresses cannot be predicted by an attacker
      without knowledge of a secret key.

   o The resulting IPv6 addresses remain stable across DHCPv6 server
      reinstallations, or even if a database of previous DHCPv6 address
      leases is not available.

   o The resulting IPv6 addresses remain stable when different DHCPv6
      servers (implementing this specification) are employed on the same
      network.

   We note that the algorithm specified in this document employs a
   (lightweight) calculated technique (as opposed to, e.g., state
   sharing among DHCPv6 servers) to achieve address stability in
   scenarios where multiple DHCPv6 servers are required to run in such a
   way as to provide increased availability, without the need of an
   additional protocol to synchronize the lease databases of DHCPv6
   servers.

   Finally, we note that the algorithm in this document is only meant to
   mitigate IPv6 address-based location tracking, device-specific
   vulnerability exploitation, and host scanning (please see [RFC7721]).
   There are a number of ways in which DHCPv6 affects user privacy,
   which the algorithm specified in this document does not mitigate (and
   does not intend to). Please see [RFC7844] for a comprehensive
   discussion of how DHCPv6 may affect user privacy.

3. Method Specification

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

   A DHCPv6 server implementing this specification must select the IPv6
   addresses to be leased with the following algorithm:

   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-256 [FIPS-SHS]. An
   implementation may provide the means for selecting other
   algorithms. Note: Message Digest 5 (MD5) [RFC1321] is
   considered unacceptable for F() [RFC6151].

Prefix:
   The prefix employed for the local subnet, as a 128-bit
   unsigned integer in network byte order (with the unused bits
   set to 0). 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.

|:
   An operator representing "concatenation".

Client_DUID:
   The DHCPv6 Unique Identifier (DUID) value contained in the
   Client Identifier option received in the DHCPv6 client
   message. The DUID can be treated as an array of 8-bit
   unsigned integers.

IAID:
   The Identity Association Identifier (IAID) value contained in
   the IA_NA option received in the client message. It must be
   interpreted as a 32-bit unsigned integer in network byte
   order.

secret_key:
   A secret key configured by the DHCPv6 server administrator,
   which must not be known by the attacker. It must be encoded
   as an array of 8-bit unsigned integers. An implementation of
   this specification must provide an interface for viewing and
   changing the secret key. All DHCPv6 servers leasing addresses
   from the same address range must employ the same secret key.

Counter:
    A 32-bit unsigned integer in network byte order that is
    employed to resolve address conflicts. It must be initialized
    to 0.

2. A candidate IPv6 address (IPV6_ADDR) to be leased is obtained by
    concatenating as many bits as necessary from the RID value
    computed in the previous step (starting from the least
    significant bit) to the Prefix employed in the equation above, as
    follows:

    IPV6_ADDR = IPV6_ADDR_LOW +
                RID % (IPV6_ADDR_HI - IPV6_ADDR_LOW + 1)

    where:

    IPV6_ADDR:
        The candidate IPv6 address to be leased.

    IPV6_ADDR_HI:
        An IPv6 address specifying the upper boundary of the IPv6
        address pool from which the DHCPv6 server leases IPv6
        addresses. If an address range is not explicitly selected,
        IPV6_ADDR_HI must be set to the IPv6 address from the Prefix
        (see the expression above) that has all of the bits of the
        Interface Identifier set to 1.

    IPV6_ADDR_LOW:
        An IPv6 address specifying the lower boundary of the IPv6
        address pool from which the DHCPv6 server leases IPv6
        addresses. If an address range is not explicitly selected,
        IPV6_ADDR_LOW must be set to the IPv6 address from the Prefix
        (see the expression above) that has all of the bits of the
        Interface Identifier set to 0.

3. The Interface Identifier of the selected IPv6 address 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 IPv6 address should be computed with
    the updated Counter value.

4. If the resulting address is not available (e.g., there is a
    conflicting binding), the DHCPv6 server should increment the
    Counter variable, and a new Interface Identifier and IPv6 address
    should be computed with the updated Counter value.

This document requires that SHA-256 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 be different for each address from a different prefix
leased to the same client. This mitigates the correlation of
activities of multihomed nodes (since each of the corresponding
addresses will employ a different Interface Identifier), host
tracking (since the network prefix, and therefore the resulting
Interface Identifier, will change as the node moves from one network
to another), and any other attacks that benefit from predictable
Interface Identifiers [RFC7721].

As required by [RFC3315], an IAID is associated with each of the
client's network interfaces and is consistent across restarts of the
DHCPv6 client.

The Counter parameter provides the means to intentionally cause this
algorithm to produce different IPv6 addresses (all other parameters
being the same). This can be of use 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 Identifiers
   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. 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, attacks that rely on predictable Interface
   Identifiers (such as [RFC7707]) are mitigated.

   The method specified in this document neither mitigates nor
   exacerbates the security considerations for DHCPv6 discussed in
   [RFC3315] and does not mitigate a range of other privacy implications
   associated with DHCPv6. Please read [RFC7844] for a comprehensive
   assessment of the privacy implications of DHCPv6.

   Finally, we note that an attacker that is able to attach to each of
   the links to which the victim attaches would still be able to
   correlate the activities of the victim across networks.

5. References

5.1. Normative References

   [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, <http://www.rfc-editor.org/info/rfc3315>.

   [RFC5453] Krishnan, S., "Reserved IPv6 Interface Identifiers",
              RFC 5453, DOI 10.17487/RFC5453, February 2009,
              <http://www.rfc-editor.org/info/rfc5453>.

   [RFC7136] Carpenter, B. and S. Jiang, "Significance of IPv6
              Interface Identifiers", RFC 7136, DOI 10.17487/RFC7136,
              February 2014, <http://www.rfc-editor.org/info/rfc7136>.

5.2. Informative References

   [FIPS-SHS]
              Federal Information Processing Standards (FIPS), "Secure
              Hash Standard (SHS)", FIPS 180-4, August 2015,
              <http://csrc.nist.gov/publications/fips/fips180-4/
              fips-180-4.pdf>.

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

   [RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321,
              DOI 10.17487/RFC1321, April 1992,
              <http://www.rfc-editor.org/info/rfc1321>.

   [RFC6151] Turner, S. and L. Chen, "Updated Security Considerations
              for the MD5 Message-Digest and the HMAC-MD5 Algorithms",
              RFC 6151, DOI 10.17487/RFC6151, March 2011,
              <http://www.rfc-editor.org/info/rfc6151>.

   [RFC7031] Mrugalski, T. and K. Kinnear, "DHCPv6 Failover
              Requirements", RFC 7031, DOI 10.17487/RFC7031, September
              2013, <http://www.rfc-editor.org/info/rfc7031>.

   [RFC7217] Gont, F., "A Method for Generating Semantically Opaque
              Interface Identifiers with IPv6 Stateless Address
              Autoconfiguration (SLAAC)", RFC 7217,
              DOI 10.17487/RFC7217, April 2014,
              <http://www.rfc-editor.org/info/rfc7217>.

   [RFC7707] Gont, F. and T. Chown, "Network Reconnaissance in IPv6
              Networks", RFC 7707, DOI 10.17487/RFC7707, March 2016,
              <http://www.rfc-editor.org/info/rfc7707>.

   [RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy
              Considerations for IPv6 Address Generation Mechanisms",
              RFC 7721, DOI 10.17487/RFC7721, March 2016,
              <http://www.rfc-editor.org/info/rfc7721>.

   [RFC7844] Huitema, C., Mrugalski, T., and S. Krishnan, "Anonymity
              Profiles for DHCP Clients", RFC 7844,
              DOI 10.17487/RFC7844, May 2016,
              <http://www.rfc-editor.org/info/rfc7844>.

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
   China

   Email: liushucheng@huawei.com