                    Privacy in IPv6 Network Prefix Assignment
                    draft-herbert-ipv6-prefix-address-privacy-00

Abstract

   This document discusses privacy concerns around network prefix
   assignment in IPv6. It evaluates the privacy threat, proposes a set
   of ideal criteria for strong privacy, and suggests solutions to
   achieve a high degree of privacy in addressing.

Status of this Memo

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

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF), its areas, and its working groups. Note that
   other groups may also distribute working documents as
   Internet-Drafts.

   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."

   The list of current Internet-Drafts can be accessed at
   http://www.ietf.org/1id-abstracts.html

   The list of Internet-Draft Shadow Directories can be accessed at
   http://www.ietf.org/shadow.html

Table of Contents

1 Introduction

   This document discusses privacy of network prefix assignment in IPv6.

   A common address assignment method is for a network to assign
   prefixes to devices. SLAAC and DHCP-PD are two mechanisms for doing
   this. In the common case of a /64 assignment (as in SLAAC) the device
   generates IIDs (interface identifiers) to create individual addresses
   within an assigned prefix. While significant effort has gone into IID
   generation techniques to protect privacy ([RFC4941], [RFC7721]), the
   privacy aspects of the prefix itself have not been fully examined.

   This document is focused on privacy within the network layer and
   specifically with privacy in addressing. There are many other privacy
   issues that arise from persistent identifiers used in higher (and
   lower) protocol layers (MAC address, session IDs, certificates,
   etc.). Discussion of these are out of scope for this document,
   however it is clear that to achieve a level of privacy that users
   deserve all layers will need to be considered.

2 The privacy concern

   In the original IPv6 addressing model, subnets (links) were assigned
   a sixty-four bit prefix [RFC4291]. Hosts in the subnet would then
   generate IIDs that are combined with the subnet prefix to create IPv6
   addresses. This model was subsequently extended to assign network
   prefixes, such as /64s, to general purpose hosts ([RFC3314],
   [RFC7934]).

   When a prefix is assigned to an end host, the prefix becomes an
   identifier for the host. So, if two such addresses have the same
   prefix (i.e. same upper sixty-four bits) then they can be assumed to
   refer to the same host. The IID portion of the addresses (lower
   sixty-four bits) are immaterial in this inference, so IID generation
   techniques don't affect the ability to make correlations.

   The fact that two addresses can be correlated to be from the same
   host implies the privacy concern. If an attacker knows that a network
   provider assigns /64 prefixes to end hosts, as is common in mobile
   networks, then it can deduce that two addresses in the provider
   prefix sharing the same sixty-four bit prefix refer to the same host.
   This correlation can be made between addresses of different flows
   independently of IIDs in those addresses. Furthermore, with a little
   more information (see Section 4.3), an attacker may not only deduce
   two addresses refer to the same end host, but also may be able to
   discover the identities of individuals in communications.

3 Prior work

   Several RFCs describe prefix assignment mechanisms and the privacy
   and security considerations for them.

3.1 SLAAC and DHCPv6-PD

   SLAAC [RFC4862] and DHCPv6-PD [RFC3633] are mechanisms to assign
   network prefixes to devices. Their respective specifications do not
   address privacy issues of prefix assignment. Security considerations
   are focused on the mechanisms.

3.2 Privacy addresses

   [RFC4941] addresses issues with persistent identifiers in IPv6. It
   describes the risks of extended use of the same identifier, and
   recommends using random interface identifiers and changing addresses
   periodically to deter inferences to reveal identify, location, or
   other privacy sensitive attributes of parties in communication.
   Addresses created by following RFC4941 recommendations are often
   called "privacy addresses".

   RFC4941 is mostly concerned with privacy and security aspects of IID
   generation. It mentions the problem of privacy of network prefixes in
   passing:

      Although it might appear that changing an address regularly in
      such environments would be desirable to lessen privacy concerns,
      it should be noted that the network prefix portion of an address
      also serves as a constant identifier. All nodes at, say, a home,
      would have the same network prefix, which identifies the
      topological location of those nodes. This has implications for
      privacy, though not at the same granularity as the concern that
      this document addresses. Specifically, all nodes within a home
      could be grouped together for the purposes of collecting
      information. If the network contains a very small number of
      nodes, say, just one, changing just the interface identifier will
      not enhance privacy at all, since the prefix serves as a constant
      identifier.

   Nevertheless, it's reasonable that some of the recommendations could
   be extrapolated to apply to prefix assignment for providing privacy.
   For instance, RFC4941 suggests to periodically do address rotation by
   generating a new IID. Conceivably, a node could periodically request
   a new network prefix via SLAAC. The new prefix would be randomized so
   that no correlation can be drawn between it and the old prefix.

As for the frequency of changing addresses, RFC4941 states:

> having large numbers of clients change their address on a daily
> or weekly basis is likely to be sufficient to alleviate most
> privacy concerns.

The statement is neither normative nor quantified. Intuitively, one
might assume that a higher frequency of address rotation reduces the
probability of privacy being compromised. However, other than the
case where a different address is used for each flow (see below),
there is no known way to quantify the relationship between frequency
of changing addresses and privacy provided to users.

A second concern with recommendations of RFC4941 is that it is was
written eleven years ago. The sophistication and capabilities of
attackers have increased substantially, so recommendations, such as
changing addresses on a daily or weekly basis, may no longer be
sufficient even if they were eleven years ago.

Presumably, one could try to achieve a high degree of privacy by
changing addresses at a high frequency (every few seconds for
instance). The effect on privacy is still unquantifiable, however
there is another problem in the disruption caused by changing
addresses. An address change would require termination of existing
flows, so a high frequency of address rotation would constantly
thrash connections. A potential mitigation would be to allow a host
to retain network prefixes for which it's still using for flows;
however, managing that would be cumbersome and likely wouldn't scale
since hosts could accumulate many prefixes over time.

The postulated exploit described in Section 4.4 would defeat the
privacy protection of any frequency of address rotation except for
the case where a different address is used per flow.

3.3 Privacy in IPv6 address generation mechanisms

[RFC7721] mainly focuses on security and privacy considerations for
IID generation. The concern around privacy in network prefix
assignment is raised:

> As [RFC4941] notes, if a very small number of nodes (say, only
> one) use a particular prefix for an extended period of time, the
> prefix itself can be used to correlate the host's activities
> regardless of how the IID is generated. For example, [RFC3314]
> recommends that prefixes be uniquely assigned to mobile handsets
> where IPv6 is used within General Packet Radio Service (GPRS).
> In cases where this advice is followed and prefixes persist for
> extended periods of time (or get reassigned to the same handsets

whenever those handsets reconnect to the same network router),
hosts' activities could be correlatable for longer periods than
the analysis below would suggest.

RFC7721 does not suggest any requirements or guidelines for privacy
in network prefixes. Similar to RFC4941, RFC7721 frames the problem
with an unquantified description as using a prefix for "extended
periods of time".

Note that RFC7721 points out that mobile handsets are often assigned
a single prefix. In this case, there is one to one relationship
between a prefix and device. For a personal device, such as a smart
phone or tablet, there would then be a one to one relationship
between a prefix and an individual user.

3.4 Host address availability recommendations

[RFC7934] recommends that general-purpose hosts are assigned multiple
globally IPv6 addresses when they attach. RFC7934 advocates prefix
assignment and /64 assignment with SLAAC in particular.

RFC7934 includes a section on host tracking (Section 9.1 of RFC7934),
however this section focuses on facilitating tracking of hosts in
provider networks to satisfy legal requirements.

From RFC7934:

  Using SLAAC with a dedicated /64 prefix for each host simplifies
  tracking, as it does not require logging every address formed by
  the host

RFC7934 references RFC4941, but does not otherwise address issues
with privacy in prefix assignment.

3.5 IPWAVE

[IPWAVE] provides the problem statement for IPWAVE. The issue of
address tracking is raised in the Security Considerations section.
From the draft:

  To prevent an adversary from tracking a vehicle by with its MAC
  address or IPv6 address, each vehicle should periodically update
  its MAC address and the corresponding IPv6 address as suggested
  in [RFC4086][RFC4941]. Such an update of the MAC and IPv6
  addresses should not interrupt the communications between a
  vehicle and an RSU.

As in the RFCs cited above, the draft suggests that addresses should

be changed periodically, however there is no guidance as to what an
acceptable frequency of change is to prevent tracking. It is
noteworthy that address change is expected to not interrupt
communications.

4 Practical effects

This section discusses the current characteristics and effects on
privacy in network prefix assignment to hosts.

4.1 Mobile networks

Privacy in prefix addressing is of particular concern in mobile
networks. It is often the case that UEs (devices such as smart
phones) are assigned a unique /64 prefix that is not shared with
other devices. As pointed out by RFC4941 and RFC7721, these network
prefixes allow the device to be tracked through correlations. For
personal devices, such as smart phones or tablets, correlations on IP
addresses could be used to infer user identities in communication.
The correlation to a user may require additional information that
might be relatively easy to acquire as demonstrated by the exploit
described in section 4.4.

Most mobile providers follow the advice of [RFC3314] and assign
single a /64 to each device. They may implement a method to force a
device to periodically request a new /64 assignment.

A sample implementation in a mobile network could assign a /64 prefix
to each IPv6 PDN, and the same prefix is retained for Idle to Active
to Idle transitions for the duration of the PDN session. If the UE is
idle without transmitting/receiving any packets, the PDN session is
dropped when the Idle Timer expires (e.g. 2 hours) and the prefix
allocation is released. So in this case the minimum amount of time
between addresses change is 2 hrs., but a device could keep its
prefix allocation indefinitely as long as the device remains active.

4.2 Connected cars

Connected cars are projected to become ubiquitous over the next
decade. By some estimates there will be 381 million connected cars on
the road by 2020, and by 2025 all new cars manufactured will be
connected. Today many vehicles are already connected to the Internet
via 4G LTE, and in the future they will connect using 5G, WiFi, DSRC
or other radio technologies. In-vehicle networks connect sensors,
displays, navigation, entertainment, as well as personal devices
being used by passengers.

Privacy in such a network is potentially a more difficult problem

since there are two independent parties that are involved in address assignment. The vehicle as a mobile node must be assigned addresses by the mobile network, and in turn the vehicle delegates addresses to devices attached to the vehicle network.

A /64 prefix could be assigned to vehicles which is a common mobile network assignment. Devices attached to the vehicle network are delegated IPv6 address within the prefix assigned to the vehicle. This results in all the attached devices sharing fate with respect to privacy. For instance, if an attacker is able to determine the location of just one device with an assigned prefix, then it can infer the location of all devices that share the same prefix. If identity of a user can be separately surmised, this raises the prospect that location of individuals can be tracked.

Periodically changing prefixes in this environment is problematic. As described in Section 3.2, a prefix change is potentially disruptive to communications as this results in an address change for each attached device. In the case of a vehicle network, the attached devices and applications they are running may be very heterogeneous such that their response and recovery for an address change may vary significantly. For instance, a laptop might attach to a vehicle network. A laptop is not normally considered a "mobile device" like a smart phone and many applications they might run don't assume addresses constantly change. Periodically changing addresses for privacy benefit may wreak havoc on such applications.

4.3 Privacy implications of NAT

Network Address Translation (NAT) is a method of remapping one IP address space into another by modifying addresses in the IP header of packets while they are in transit across a routing node. NAT has been extensively deployed to allow hosts that are assigned IPv4 private addresses [RFC1918] to communicate with hosts in the global Internet. NAT has been used to extend the usefulness of IPv4 in the face of address depletion.

A side effect of NAT (possibly accidental) is that NAT modifies addresses such that it obfuscates the identity of the source host behind a NAT. With a significant population of users sharing a pool of NAT addresses, an external observer can draw little correlation based on addresses between flows that have gone through a NAT device. The result is that NAT provides strong privacy in addressing. NAT use is of particular concern to law enforcement since its privacy characteristics complicate criminal investigation [EUROPOL].

4.4 Exploit to defeat prefix rotation

As mentioned in a Section 3.2, one might try to provide privacy in
addressing by changing addresses with a high frequency. The following
exploit is postulated as a way to defeat the privacy goals of
periodic address rotation at any frequency except when a different
address is used for each connection.

The exploit is:

> o An attacker creates an "always connected" app that provides some
>   seemingly benign service and users download the app.

> o The app includes some sort of persistent identity. For instance,
>   this could be an account login.

> o The backend server for the app logs the identity and IP address
>   of a user each time they connect.

> o When an address change happens, existing connections on the user
>   device are disconnected. The app will receive a notification and
>   immediately attempt to reconnect using the new source address.

> o The backend server will see the new connection and log the new
>   IP address as being associated to the user. Thus, the server has
>   a real-time record of users and the IP address they are using.

> o The attacker intercepts packets at some point in the Internet.
>   The addresses in the captured packets can be time correlated
>   with the server database to deduce identities of parties in
>   communications that are unrelated to the app.

5 Criteria for strong privacy

A set of "ideal" criteria for strong privacy in addressing can be
established. These criteria are intended to be specific, such that
when applied to a solution the amount of information that can be
inferred by correlating addresses is quantifiable.

The ideal criteria for IPv6 addresses that provide strong privacy
are:

> o Addresses are composed of a global routing prefix and a suffix
>   that is internal to an organization or provider. This is the
>   same property for IP addresses [RFC4291].

> o The registry and organization of an address can be determined by
>   the network prefix. This is true for any global address. The
>   organizational bits in the address should have minimal hierarchy
>   to prevent inference. It might be reasonable to have an internal

prefix that divides identifiers based on broad geographic
regions, but detailed information such as location, department
in an enterprise, or device type should not be encoded in a
globally visible address.

o Given two addresses and no other information, the desired
properties of correlating them are:

o It can be inferred if they belong to the same organization and
registry. This is true for any two global IP addresses.

o It may be inferred that they belong to the same broad
grouping, such as a geographic region, if the information is
encoded in the organizational bits of the address.

o No other correlation can be established. It cannot be inferred
that the IP addresses address the same node, the addressed
nodes reside in the same subnet, rack, or department, or that
the nodes for the two addresses have any geographic proximity
to one another.

Note that if NAT is deployed with a sufficiently large population of
users sharing a pool of IP addresses then these criteria are met.
Thus NAT can be considered a baseline for strong privacy in
addressing.

6 Identifier/locator split solution

This section proposes using identifier/locator split to meet the
strong privacy criteria for addressing in IPv6.

6.1 Overview

Identifier/locator split separates the notions of location and
identity in IP addresses. Identifier addresses are addresses that
don't contain topological information for routing within a network.
Nodes are assigned identifier addresses that can be used as endpoints
in communications. Locator addresses indicate the topological
location of a logical node. In order to forward a packet to a
destination with an identifier address, an ingress node for a network
maps an identifier address to a locator address. A network overlay
method is used to forward the packet to the location in the network
of the logical or mobile node.

Since identifier addresses are non-topological they don't require any
hierarchy in address assignment beyond the global network prefix.
Therefore the network can randomly generate identifier addresses
within a portion of the address in a space of at least sixty-four

bits.

Strong privacy in addressing can be achieved by using a different
randomly generated identifier address for each flow. Conceptually,
this would entail that the network creates and assigns a unique and
untrackable address to a host for every flow created by a host. Some
suggestions for scaling this technique are discussed below.

Note that this technique parallels what NAT does in that NAT
effectively creates a different source address per connection. Unlike
NAT however, address assignments in identifier/locator split are
stateless in the network and transparent to the end points.

6.2 Scaling identifier/locator address assignment

Assigning an address per connection is a potential scaling problem on
two accounts:

    o The amount of state needed in the mapping system is significant.

    o Bulk host address assignment is inefficient.

6.2.1 Scaling the amount of mapping state

The amount of state necessary to assign each flow its own unique
source IP address is equivalent, or at least proportional, to the
amount of state needed for NAT-- basically this is one state element
for every connection in the network. So in one sense this solution
should scale as well as NAT has.

6.2.1.1 Hybrid address assignment

Not all communications might require strong privacy, so it is
conceivable that a hybrid approach to address assignment might be
taken. A network might assign prefixes for use with communications
that are not privacy sensitive, and may assign singleton addresses
that meet strong privacy criteria for privacy sensitive
communications. Assuming that most communications don't need strong
privacy this could reduce the amount of state needed in the mapping
system considerably. The decision as to whether strong privacy is
required for a communication would be made by the user or
application.

6.2.1.2 Hidden aggregation

A possible solution to reduce state is to make addresses aggregable,
but use an aggregation method that is known only by the network
provider and hidden to the rest of the world. The network could use a

reversible hash or encryption function to create addresses.

The input to an address generation function includes a device identifier, a secret key, and a generation index.

The function may have the form:

    Address = Func(key, dev_ident, gen)

Where "key" is secret to network, "dev_ident" is a network internal identifier for a device (roughly equivalent to "identity" in IDEAS), and "gen" is generation number 0,1,2,... N. The generation value is changed for each invocation to create different addresses for assignment to a device.

When a network ingress node is forwarded a packet it performs the inverse function on an address.

The inverse function has the form:

    (dev_ident, gen) = FuncInv(key, Address)

The returned dev_ident value is used as the identifier in the mapping lookup for a locator address. In this manner, the network can generate many addresses to assign to a device where they all share a single entry in the mapping system.

6.2.2 Scaling bulk address assignment

Assigning multiple addresses without aggregation is difficult to scale. Each address would need to be individually specified in an assignment sent to a host.

6.2.2.1 Bulk assignment using DHCPv6

DHCPv6 might allow bulk singleton address assignment. As stated in [RFC7934]:

        Most DHCPv6 clients only ask for one non-temporary address, but
        the protocol allows requesting multiple temporary and even
        multiple non- temporary addresses, and the server could choose
        to provide multiple addresses. It is also technically possible
        for a client to request additional addresses using a different
        DHCP Unique Identifier (DUID), though the DHCPv6 specification
        implies that this is not expected behavior ([RFC3315], Section
        9). The DHCPv6 server will decide whether to grant or reject
        the request based on information about the client, including its
        DUID, MAC address, and more. The maximum number of IPv6

addresses that can be provided in a single DHCPv6 packet, given
a typical MTU of 1500 bytes or smaller, is approximately 30.

6.2.2.2 Hidden aggregation assignment

By extending the concept of hidden aggregation assignment (section
6.2.1.2), it is conceptually possible that a host could work in
concert with the network to generate addresses that meet strong
privacy criteria. In this method, a host autonomously generates
addresses as needed. The network, but no one outside the network, is
then able aggregate the addresses as belonging to the device.

End hosts are generally considered untrusted nodes by the network, so
they cannot be given access to the network secret key used for the
address generation function. Public key encryption might be used.

A host may perform an encryption function to generate addresses:

    Address = Encrypt(pub_key, dev_inet, gen)

Where "pub_key" is a public key for the network, "dev_ident" is a
network identifier for the device and is visibile to the device (so
it may be leaked). "gen" is a generation number 0,1,2,... N. The
generation value is changed for each invocation to create different
addresses.

When a network ingress node is forwarded a packet it decrypts an
address using the network private key.

The decryption function has the form:

    (dev_ident, gen) = decrypt(priv_key, Address)

Where "priv_key" is the secret private key of the network associated
with the public key. The returned dev_ident value is used as the
identifier in the mapping lookup for a locator address.

Note that this method would require an new address assignment
protocol.

6.2.3 Practicality of hidden aggregation methods

The premise of hidden aggregation is that only trusted devices in the
network are able decode the aggregation hidden within IPv6 addresses.
This implies that the network must keep secrets about the process. In
the above examples, the secrets are keys used in the hash or
encryption. The security of the key is then paramount, so techniques
for key management, rotation, and using different key sets for

obfuscation are pertinent.

To perform a mapping lookup a node must apply the inverse address generation function to map addresses to locators. This lookup would occur in the critical data path so performance is important. Encryption and hashing are notoriously time consuming and computationally complex functions.

Some possible mitigating factors for performance impact are:

o The input to address generation functions is a small amount of data and has fixed size. The input is a key (presumably 128 or 256 bits), part of all of an IPv6 address (128 bits), and a generation number (sixteen to twenty-four bits should work).

o Given that the input is fixed size, specialized hardware might be used to optimize performance of the inverse address generation function. For instance, modern CPUs include instructions to perform crypto [AES-NI]. Since the keys used in these functions are secret to the network and there are relatively few of them, they might be preloaded into a crypto engine to reduce setup costs.

o The output of an inverse address generation function is cacheable. A cache on a device could contain address to locator mappings. When the inverse function and lookup on dev_ident are performed, a mapping of address to the discovered locator could be created in the cache. The device could then map addresses in subsequent packets sent on the same flow to the proper locator by looking up the address in the cache.

6.3 Law enforcement considerations

This section discusses law enforcement considerations for host tracking when using an identifier/locator split solution for strong privacy. NAT is used as a reference point for discussion.

There are two sub-problems expressed by law enforcement about NAT [EUROPOL]:

1) It is difficult to map a NAT address and port back to a user.

2) Many Internet servers do not log the client source port of connections.

The first problem is one of maintaining a log of NAT mappings. If the log contains the inner address, outer address and port, and timestamp when the NAT mapping was created-- then given the log and a NATed

packet, the original sender can be revealed. Note that NAT logs are kept internal to the provider network, and securing them is the responsibility of the provider. The same model can be applied to identifier/locator split where the infrastructure keeps a log of identifier to locator mappings and a timestamp for when they were created.

In the second problem, the source port is needed to be logged in servers in order correlate a flow to an entry in the NAT logs of a provider. The source port is relevant to a NAT mapping; however, in identifier/locator split it's not since identification of a host node contained with an address. Therefore the client source port is not required for tracking users in an identifier/locator solution.

7 Security considerations

The subject of this draft is privacy assigning network prefixes. Implicit to this is that any address assignment technique requires security on the parties entities involved.

In the identifier/locator split the mapping of identifier to locator is privacy sensitive information. The locator may very well imply the geo location of a device. As such, it is recommended that locators that might contain accurate location information are strictly contained within a trusted infrastructure.

In mobile environments, it is natural to group identifiers (addresses) together that have the same attributes [IDGROUP]. For instance, if as in section 6.1 a different source address is used for each flow, all of the addresses assigned to a device form a group. When the device moves, all of the addresses move with it; this can be efficiently implemented as single operation on the mapping system. The group information is thus privacy sensitive information that must be secured by the infrastructure to prevent use of the information to make inferences of identity similar to /64 assignment.

Hidden aggregation is a means of grouping identifiers together similar to the above description. The secret keys used in these algorithms are thus critical information that must be kept secure. Security by obscurity should be avoided here, divulging the algorithm used to generate addresses should not reduce security or privacy.

End hosts must implement appropriate security to ensure privacy. For instance, if an address is assigned per flow as described in Section 6.2, applications must be isolated from one another so that they cannot infer addresses or privacy properties of other applications running within the same system. Also, if a host is completely compromised then that fact should not impact the privacy and security

of other hosts and applications within a network.

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/info/rfc2119>.

   [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6
            (IPv6) Specification", STD 86, RFC 8200, DOI
            10.17487/RFC8200, July 2017, <https://www.rfc-
            editor.org/info/rfc8200>.

8.2 Informative References

   [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy
            Extensions for Stateless Address Autoconfiguration in
            IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007,
            <https://www.rfc-editor.org/info/rfc4941>.

   [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, <https://www.rfc-
            editor.org/info/rfc7721>.

   [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing
            Architecture", RFC 4291, DOI 10.17487/RFC4291, February
            2006, <https://www.rfc-editor.org/info/rfc4291>.

   [RFC3314] Wasserman, M., Ed., "Recommendations for IPv6 in Third
            Generation Partnership Project (3GPP) Standards", RFC 3314,
            DOI 10.17487/RFC3314, September 2002, <https://www.rfc-
            editor.org/info/rfc3314>.

   [RFC7934] Colitti, L., Cerf, V., Cheshire, S., and D. Schinazi, "Host
            Address Availability Recommendations", BCP 204, RFC 7934,
            DOI 10.17487/RFC7934, July 2016, <https://www.rfc-
            editor.org/info/rfc7934>.

   [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>.

   [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic
             Host Configuration Protocol (DHCP) version 6", RFC 3633,
             DOI 10.17487/RFC3633, December 2003, <https://www.rfc-
             editor.org/info/rfc3633>.

   [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/info/rfc3315>.

   [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G.,
             and E. Lear, "Address Allocation for Private Internets",
             BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996,
             <https://www.rfc-editor.org/info/rfc1918>.

   [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/info/rfc3315>.

   [EUROPOL] EUROPOL/EC3 to delegations of the Council of the European
             Union, "Carrier-Grade Network Address Translation (CGN) and
             the Going Dark Problem", January 2017

   [AES-NI] Gueron, S., "Intel Advanced Encryption Standard (AES) New
             Instructions", <https://software.intel.com/sites/
             default/files/article/165683/aes-wp-2012-09-22-v01.pdf>

   [IDGROUP] Herbert, T., "Identifier groups", draft-herbert-idgroups-
             00

9 Acknowledgments

   The author would like to thank Robert Moskowitz for insightful
   comments and contributions to this draft.

Authors' Addresses

   Tom Herbert
   Quantonium
   Santa Clara, CA
   USA

   Email: tom@quantonium.net