Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: November 28, 2018

D. O'Reilly
May 27, 2018

A Model for Storing IPv6 Stateless Address Autoconfiguration Crime Attribution Records in a Privacy Sensitive Way
draft-daveor-slaac-privacy-logging-00

Abstract

The need for individual right to privacy and the need for law enforcement to be able to effectively investigate crime are sometimes portrayed as being irreconcilably in direct conflict with each other. Both needs are legitimate and ignoring the challenges presented by areas of conflict will not make the problem go away.

The document presents a conceptual model that allows for both sets of requirements to be met simultaneously. The reason for this publication is to show that, with some creative thinking, it is possible to identify win-win solutions that simultaneously achieve both privacy and law enforcement goals.

Table of Contents

1. Introduction

IPv6 addresses are assigned to organisations in blocks that are much larger than the size of the blocks in which IPv4 addresses are assigned, with common IPv6 prefix sizes being /48, /56 and /64 [RFC6177], [RIPE 699]. Current regulatory models typically oblige ISPs to keep records to facilitate identification of their subscribers, and in the case of IPv6 this will mean recording the prefix(es) have been assigned to each customer.

From the perspective of crime attribution, therefore, when a specific IP address is suspected to be associated with criminal activity, records will most likely available from an ISP to identify the organisation to which the prefix has been assigned. The question

then arises how an organisation approached by law enforcement authorities, particularly a large organisation, would be able to ascertain which host/endpoint within their network was using a particular IP address at a particular time.

This is not a new problem, with many difficulties of crime attribution already present in the IPv4 Internet. Nevertheless, it is worthwhile to consider the crime attribution characteristics of IPv6 in anticipation of wider deployment of this technology in the coming years.

IPv6 provides several mechanisms through which hosts can be assigned an IP address. [RFC7721] provides a list of these. Briefly they can be summarised as:

- Manually configured addresses
- DHCPv6 assigned addresses
- Stateless Address Autoconfiguration (SLAAC)
- Addresses derived from an IPv4 address (transitional)

When approached by a law enforcement agency to identify the host/endpoint that was using a particular IP address at a particular time, the organisation's ability to deliver this information will depend on how IPv6 addresses are being assigned to endpoints within their network.

1.1. SLAAC: Stateless Address Autoconfiguration

IPv6 Stateless Address Autoconfiguration (SLAAC) describes the process used by a host in deciding how to auto configure its interfaces in IPv6[RFC4862]. This includes generating a link-local address, generating global addresses via stateless address autoconfiguration and then using duplicate address detection to verify the uniqueness of the addresses on the link. SLAAC requires no manual configuration of hosts, minimal (if any) configuration of routers, and no additional servers.

Routers advertise prefixes that identify the subnet(s) associated with a link and hosts generate an interface identifier that uniquely identifies an interface on a subnet. An address is formed by combining these two. In the absence of a router, hosts generate only link-local addresses. Autoconfiguration is only possible on multicast-capable links.

The process begins by generating a link-local address for the interface. This is achieved by appending the interface identifier to the well-known link-local prefix. At this point, the address is considered "tentative" because it might be in use by another host on the network. The host verifies the uniqueness of the address by sending a Neighbour Solicitation message containing the tentative address. If the address is already in use, the node that is using that address will send back a Neighbour Advertisement message. If the address is not unique, auto configuration stops and manual configuration is required or an alternative interface identifier can be used, if one is configured.

Once it has been established that the link-local address is unique, it is assigned to the interface. Next, the host listens for a Router Advertisement message or, if the host does not want to wait, it can send a Router Solicitation message to the all-routers multicast group.

Router Advertisement messages contain zero or more prefix information options that contain information that can be used to generate global addresses. Hosts can use stateless address autoconfiguration and DHCPv6 simultaneously if they want. If the Router Advertisement indicates that the prefixes can be used for autoconfiguration (by setting the "autonomous address-configuration flag" in the Prefix Information option field) it will also include a subnet prefix and lifetime values, indicating how long addresses created from this prefix will remain preferred and valid. Hosts process all Router Advertisements that are received periodically, adding to and refreshing the information received in previous advertisements.

Crucial to the crime attribution properties of SLAAC is the selection of interface identifier. Various algorithms exist for the generation of interface identifiers, depending on whether the interface identifier is intended to be stable (long-lived) or temporary. The following two sub-sections describe stable and temporary interface identifier generation algorithms.

1.1.1. Stable Address Autoconfiguration

Originally, various standards specified that the interface identifier should be generated from the link-layer address of the interface. For example [RFC2467], [RFC2470], [RFC2491], [RFC2492], [RFC2497], [RFC2590], [RFC3164], [RFC3527], [RFC4338], [RFC4391], [RFC5072], [RFC5121]. This is used in cases where a stable IPv6 address is being generated.

[RFC8064] changes the recommended default interface identifier generation scheme when SLAAC is in use to generate stable IPv6

addresses. It recommends against embedding stable link-layer addresses in IPv6 interface identifiers, recommending instead the use of a semantically opaque value as defined in [RFC7217] over all other alternatives. [RFC8064] also highlights some reasons why a stable IPv6 address would be desirable. For example, network management, event logging, enforcement of access control, provision of quality of service or for server or router interfaces. Similarly, they allow long-lived TCP connections. However, the document does not make recommendations about WHEN stable addresses should be used and when temporary addresses should be used.

[RFC7271] describes a method where an IPv6 address can be configured in such a way that it is stable within each subnet but the interface identifier changes when the host moves from one network to another. In general terms, the approach is to pass the following values to a cryptographic hash function (such as SHA1 or SHA256):

- The network prefix
- The network interface id
- The network id (subnet, SSID or similar) - optional parameter
- A duplicate address detection counter - incremented in case of a duplicate address being generated
- A secret key (128 bits long at least)

The interface identifier is generated by taking as many bits, starting at the least significant, as required. The result is an opaque bit stream that can be used as the interface id.

1.1.2. Temporary Address Autoconfiguration

[RFC4941] describes a system by which interface identifiers generated from an IEEE identifier (EUI-64) can be changed over time, even in cases where the interface contains an embedded IEEE identifier. These are referred to as temporary addresses.

The reason behind development of this technique is that the use of a globally unique, non-changing, interface identifier means that the activity of a specific interface can be tracked even if the network prefix changes. The use of a fixed identifier in multiple contexts allows correlation of seemingly unrelated activity using the identifier. Contrast this with IPv4 addresses, where if a person changes to a different network their entire IP address will change.

The goals of the temporary address generation procedure are that:

- Temporary address generation does not result in any changes to the basic behaviour of addresses generated via SLAAC.

- The temporary address generation algorithm creates additional addresses based on a random interface identifier for the purpose of initiating outgoing sessions. These temporary addresses would be used for a short period of time (hours to days) and would then be deprecated.

- The algorithm produces a sequence of temporary global scope addresses from the sequence of interface identifiers that appear to be random in the sense that it is difficult for an outside observer to predict a future address (or identifier) based on the current one, or to determine a previous address (or identifier) knowing only the current one.

- By default, the algorithm generates a set of addresses from the same interface identifier, one for each prefix for which a global address has been generated via SLAAC. Using the same interface identifier to generate a set of temporary addresses reduces the number of IP multicast groups that a host must join.

- A node highly concerned about privacy may use different identifiers for different prefixes, resulting in a set of global addresses that cannot be easily tied to each other.

To prevent the generation of predictable values, the algorithm must contain an unpredictable component. The algorithm assumes that each interface maintains an associated randomised interface identifier. When temporary addresses are generated, the current value of the interface identifier is used. The algorithm also assumes that for a given temporary address, the implementation can determine the prefix from which it was generated.

Two approaches to generate random interface identifiers are presented in [RFC4941], depending on whether stable storage is present.

When stable storage is present, it is assumed that a 64-bit history value is available and can be used. This value is generated as described below. The first time the system boots, a random value is selected.

1. Take the history value and append it to the interface identifier generated as described in [RFC4291].

2. Compute the MD5 of the resulting value

3. Take the leftmost 64 bits and set bit 6 to zero. (This creates an interface identifier with the universal/local bit indicating local significance only)

4. Compare the generated identifier against a list of reserved identifiers and to those already assigned to an address on the local device. If an unacceptable value has been generated, start again at step 1.

5. Save the generated identifier.

6. Take the rightmost 64 bits and save them to state storage as the history value for the next iteration of the algorithm.

When stable storage is not present, no history value will be available. Therefore, the initial history value should be generated at random. Algorithms other than MD5 can be used to compute the temporary address if desired.

Other approaches such as cryptographically generated addresses (CGA) can be used to generate random interface identifiers based on the public key of the node [RFC3972]. The goal of CGAs is to prove ownership of an address and prevent spoofing and stealing of IPv6 addresses. The CGA process may not be suitable for privacy addresses because (a) it requires nodes to have a public key, meaning the node can be identified by the key and (b) it is computationally intensive, discouraging frequent regeneration.

Devices implementing this specification must provide a way for end users to explicitly enable or disable the use of temporary addresses. Also, sites might wish to disable it, so implementations should provide a way for trusted system administrators to enable or disable the use of temporary addresses. Implementations should also provide a way to enable and disable generation of temporary addresses for specific prefix subranges.

1.1.3. Crime Attribution Characteristics

1.1.3.1. Stateless Address Autoconfiguration

IPv6 addresses are assigned to organisations in blocks much larger than the size of the blocks in which IPv4 addresses are assigned. The question arises about how an organisation approached by law enforcement authorities, particularly a large organisation, will be able to ascertain which host/endpoint within their organisation was using a particular IP address at a particular time when addresses have been assigned using SLAAC.

From the crime attribution perspective, both the recommended stable and temporary address generation algorithms pseudo-randomly select addresses from the space of available addresses. When SLAAC is being used, the hosts auto-configure the IP addresses of their interfaces, meaning there is no organisational record of the IP addresses that have been selected by particular hosts at particular points in time.

1.1.3.2. SLAAC with stable interface identifiers

From a crime attribution point of view, the use of a stable interface identifier (whether generated for a link-local address or otherwise) will provide some measure of assurance that it will be possible to identify a specific host/interface based on the IPv6 address. While it may not be possible for a network administrator to calculate the interface identifier (and therefore the IPv6 address) that will be used by a specific interface, due to the presence of a secret key, with some effort it should be possible for a network operator to determine which host/endpoint, or at least a relatively small subset of hosts/endpoints, is responsible for traffic arising from a particular IPv6 address.

Due to the relatively long-term use of a particular address by an interface, it is at least possible that an organisation might be able to use traffic flow analysis or other similar network monitoring techniques to identify the endpoint using the address. This assumes that the IPv6 address is still active and generating traffic. It will also, of course, only identify the endpoint using the address at the time of the traffic flow analysis and not at the time of the alleged criminal activity that is under investigation.

1.1.3.3. SLAAC with temporary interface identifiers

The problem of crime attribution is exacerbated in the case of temporary interface identifier generation due to the fact that the generated addresses are the endpoint's preferred IPv6 address, by default, for a period of one day [RFC4941].

It is difficult to see how the activity of IPv6 addresses generated using temporary interface identifiers could be attributed to any host/endpoint. The interface identifier generation algorithm has a cryptographic component, meaning that the addresses will appear to be pseudo-randomly selected from the range of available addresses.

Even presuming that the host/endpoint is still active and generating traffic there is no apparent way to associate the activity of the host/endpoint's current address with the address in use at the time of the alleged criminal activity.

This attribution problem is "by design", arising from the expected behaviour of SLAAC with temporary interface identifiers. It therefore seems that the crime attribution challenges that will arise from the use of this technology have not been given due consideration. The use of this technology will likely become a significant crime attribution challenge in future.

2. Scope

This document presents a record-retention model whereby it is possible for an organisation, if required to do so as part of a criminal investigation, to answer the question "Who was using IP address A at a particular point in time?" without being able to answer any more broadly scoped questions, such as "What were all of the IP addresses used by a particular person?"

3. Model

3.1. Assumptions

The model described here makes the following assumption:

- The endpoint/interface for which the IPv6 address is being generated has a meaningful, unique identifying characteristic. Whether that is the layer two address of the interface or some other organisational characteristic is unimportant for the purpose of the model.

3.2. Record Generation

The host generates a temporary IPv6 address using any of the techniques described above, but most likely the technique described in [RFC4941]. Having completed the duplicate address detection phase of SLAAC, but before beginning to use the IP address for communication, the host creates a structure of the following form:

```
typedef struct {
        const char *LOG_ENTRY_TAG="__LOG_ENTRY_TAG__";
        unsigned char *ip_address;
        unsigned int identifying_characteristic_length;
        unsigned char *identifying_characteristic;
        unsigned int client_generation_time;
        unsigned int client_preferred_time;
        unsigned int client_valid_time;
} log_entry;
```

The fields in the structure are all mandatory, and populated as follows:

- LOG_ENTRY_TAG has the fixed, constant value "__LOG_ENTRY_TAG__"
- ip_address contains the 16 byte IPv6 address
- identifying_characteristic_length contains the byte length of the identifying_characteristic field
- identifying_characteristic is a variable length byte string, organisationally interpreted, to represent the identifying characteristic of the host generating the IPv6 address
- client_generation_time contains the time, in seconds since the unix epoch, as recorded by the client creating the IPv6 address, at which the address was generated
- client_preferred_time contains the period, in seconds, starting at client_generation_time for which the client will use this IPv6 address as its preferred address
- client_valid_time contains the period, in seconds, starting at client_generation_time for which the client will consider this IPv6 address to be valid

When the structure has been populated, the host encrypts the structure using AES-128 in CBC mode with the selected IPv6 address being used as the encryption key.

The record message is now ready for transmission.

3.3. Record Transmission and Storage

The host submits the completed record to a specified multicast address and port but, when sending the record, sends it using the unspecified IPv6 address (i.e. "::") as the source IP address.

When records are received by a logging server that is listening to the specified multicast address, the logging server creates a new log entry consisting of:

- The time the record was received, ideally calibrated to a global standard time (e.g. NTP) with the granularity of a second.
- The received encrypted record as a binary blob.

3.4. Record Querying

If and when it becomes necessary to query the recorded entries, the following (representative) process can be followed:

1. Taking the IP address for which the attribution information is required, iterate through all recorded log entries and use the IP address as a decryption key and attempt to decrypt the record.

2. Examine the decrypted data and check whether the first 17 bytes have the values "__LOG_ENTRY_TAG__".

    A. If so:

        i This indicates that the log entry has been successfully decrypted.

        ii The IP address contained in the log entry can be verified against the IP address that was used as a key to confirm that the log entry contains the correct value.

        iii The identifying characteristic can then be read from the log entry, along with the time at which the host generated the IP address.

        iv The client_preferred_time and client_valid_time fields can be used to check whether the IPv6 address was valid and/or preferred by the client at the time of interest.

        v The time in the record can be correlated with the time in the log entry recorded by the server so that any time differential can be compensated for.

    B. If not:

        i This indicates that the log entry has not been successfully decrypted and that the current log entry pertains to a different IP address.

        ii Move on to the next log entry and try again.

As described in the next section, it would be computationally feasible to use this process on a large number of log entries but, if necessary, the space of log entries to be searched can be reduced by selecting a range of log entries based on the time recorded by the server.

4. Proof of Concept

A proof of concept implementation of the model above has been developed. Log entries using pseudorandom IPv6 addresses were generated for a network of 20,000 computers, changing IP address every day (which is the default specified in [RFC4941]) for two years. This leads to the generation of 14.6 million log entries.

Code was developed to select a random IP address, known to be represented in the log entries, and search the entire log for entries that are successfully decrypted using that IP address. This code was executed 10,000 times and the following results were noted:

1. On a single CPU PC with an Intel Core i7 running at 2.8GHz it takes on average 11.34 seconds (standard deviation 0.82 seconds) to check the entire log.

2. In all 10,000 test cases, only a single log entry was successfully decrypted - the one representing the attribution data for the target IP address.

5. IANA Considerations

This memo includes no request to IANA.

6. Security Considerations

6.1. Cryptographic Strength

The strength of the key comes from the length and pseudo-random nature of the IPv6 address generation mechanism, the very feature that is desirable from a privacy perspective.

In order to decrypt a specific log entry without knowing the target IP address, a brute force approach must be adopted. Presuming a known 64-bit address prefix, means that there is a space of 2^64 possible addresses to search.

Code was also developed to attempt to brute force log entries, and it was noted that on the same PC used for the testing above (single CPU PC with an Intel Core i7 running at 2.8GHz) attempting to brute force a single log entry would be computationally infeasible (approximately 22,313,257 years required). To decrypt the entire log would require this same amount of time for each individual log entry.

6.2. Injection of False Records

In the model presented here, there is no mechanism to detect injection of false records. A shared secret cryptographic model could be developed but in order to maintain the privacy characteristics of the concept, all authorised endpoints would need to use the same shared secret otherwise it would be possible to a rogue log recorder to reduce the range of possible hosts through correlation of the encryption key.

6.3. Retention Period of Records

The period of time for which logs should be retained is, broadly speaking, out of scope of this discussion.

Depending on national legislation there will be obligations on certain types of organisations to retain logs for particular periods of time. Most other organisations do not have any legal obligation to retain records of which endpoint was using a specific IP address at a particular point in time, although these records are often kept for other reasons such as network security, performance monitoring and troubleshooting.

7. Conclusion

The model presented here provides a balance between the needs for individual privacy at the network layer while also providing a mechanism for recording data that would be required in a criminal investigation. The balance that has been proposed here is at the point where it is possible to identify, using this technique, who was using a specific IP address at a specific point in time without being able to extract any more information such as all of the people who were using a particular IP or all of the IP addresses that were used by a particular endpoint.

8. Normative References

[RFC2467] Crawford, M., "Transmission of IPv6 Packets over FDDI Networks", RFC 2467, DOI 10.17487/RFC2467, December 1998, <https://www.rfc-editor.org/info/rfc2467>.

[RFC2470] Crawford, M., Narten, T., and S. Thomas, "Transmission of IPv6 Packets over Token Ring Networks", RFC 2470, DOI 10.17487/RFC2470, December 1998, <https://www.rfc-editor.org/info/rfc2470>.

[RFC2491] Armitage, G., Schulter, P., Jork, M., and G. Harter, "IPv6 over Non-Broadcast Multiple Access (NBMA) networks", RFC 2491, DOI 10.17487/RFC2491, January 1999, <https://www.rfc-editor.org/info/rfc2491>.

[RFC2492] Armitage, G., Schulter, P., and M. Jork, "IPv6 over ATM Networks", RFC 2492, DOI 10.17487/RFC2492, January 1999, <https://www.rfc-editor.org/info/rfc2492>.

[RFC2497] Souvatzis, I., "Transmission of IPv6 Packets over ARCnet Networks", RFC 2497, DOI 10.17487/RFC2497, January 1999, <https://www.rfc-editor.org/info/rfc2497>.

[RFC2590] Conta, A., Malis, A., and M. Mueller, "Transmission of IPv6 Packets over Frame Relay Networks Specification", RFC 2590, DOI 10.17487/RFC2590, May 1999, <https://www.rfc-editor.org/info/rfc2590>.

[RFC3164] Lonvick, C., "The BSD Syslog Protocol", RFC 3164, DOI 10.17487/RFC3164, August 2001, <https://www.rfc-editor.org/info/rfc3164>.

[RFC3527] Kinnear, K., Stapp, M., Johnson, R., and J. Kumarasamy, "Link Selection sub-option for the Relay Agent Information Option for DHCPv4", RFC 3527, DOI 10.17487/RFC3527, April 2003, <https://www.rfc-editor.org/info/rfc3527>.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <https://www.rfc-editor.org/info/rfc3972>.

[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>.

[RFC4338] DeSanti, C., Carlson, C., and R. Nixon, "Transmission of IPv6, IPv4, and Address Resolution Protocol (ARP) Packets over Fibre Channel", RFC 4338, DOI 10.17487/RFC4338, January 2006, <https://www.rfc-editor.org/info/rfc4338>.

[RFC4391] Chu, J. and V. Kashyap, "Transmission of IP over InfiniBand (IPoIB)", RFC 4391, DOI 10.17487/RFC4391, April 2006, <https://www.rfc-editor.org/info/rfc4391>.

[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>.

[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>.

[RFC5072] Varada, S., Ed., Haskins, D., and E. Allen, "IP Version 6 over PPP", RFC 5072, DOI 10.17487/RFC5072, September 2007, <https://www.rfc-editor.org/info/rfc5072>.

[RFC5121] Patil, B., Xia, F., Sarikaya, B., Choi, JH., and S. Madanapalli, "Transmission of IPv6 via the IPv6 Convergence Sublayer over IEEE 802.16 Networks", RFC 5121, DOI 10.17487/RFC5121, February 2008, <https://www.rfc-editor.org/info/rfc5121>.

[RFC6177] Narten, T., Huston, G., and L. Roberts, "IPv6 Address Assignment to End Sites", BCP 157, RFC 6177, DOI 10.17487/RFC6177, March 2011, <https://www.rfc-editor.org/info/rfc6177>.

[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, <https://www.rfc-editor.org/info/rfc7217>.

[RFC7271] Ryoo, J., Ed., Gray, E., Ed., van Helvoort, H., D'Alessandro, A., Cheung, T., and E. Osborne, "MPLS Transport Profile (MPLS-TP) Linear Protection to Match the Operational Expectations of Synchronous Digital Hierarchy, Optical Transport Network, and Ethernet Transport Network Operators", RFC 7271, DOI 10.17487/RFC7271, June 2014, <https://www.rfc-editor.org/info/rfc7271>.

[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>.

[RFC8064] Gont, F., Cooper, A., Thaler, D., and W. Liu, "Recommendation on Stable IPv6 Interface Identifiers", RFC 8064, DOI 10.17487/RFC8064, February 2017, <https://www.rfc-editor.org/info/rfc8064>.

[RIPE_699] RIPE, "IPv6 Address Allocation and Assignment Policy", 2016, <https://www.ripe.net/publications/docs/ripe-699>.

Author's Address

David O'Reilly
Ireland

Email: rfc@daveor.com