IPv6 Ephemeral Addresses

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of 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.

Abstract

This document describes a new address type that is called "Ephemeral Addresses". Ephemeral Addresses are designed to be used as clients’ source addresses of TCP / UDP sessions. An idea Ephemeral Addresses is simple enough. They are achieved by deriving existing "ephemeral ports" specifications. In other words, they are achieved by naturally upgrading their concept from the port space to the address space. Since Ephemeral Addresses functions are implemented only in the kernel side of the OS, we can use the Ephemeral Addresses functions in current exiting enormous client applications without modifying them. Ephemeral Addresses functions can contribute to various types of security enhancements that include privacy protections etc.
1. Introduction

In current communication style, IP communication sessions are multiplexed at two different layers (Network and Transport). In the IPv4 era when one node owns one IP address, this communication style was reasonable. However, we are moving to the IPv6 era that it has become normal for one node to own multiple IP addresses. This communication style is getting less optimized. It must be time to reconsider current communication style and to find suitable communication style for the IPv6 era.

As a first step, this document proposes a new address type that is called "Ephemeral Addresses". Ephemeral Addresses are designed to be used as clients’ source addresses of TCP / UDP sessions. An idea Ephemeral Addresses is simple enough. They are achieved by deriving existing "ephemeral ports" specifications. In other words, they are achieved by naturally upgrading their concept from the port space to the address space. Since Ephemeral Addresses functions are implemented only in the kernel side of the OS, we can use the Ephemeral Addresses functions in current exiting enormous client applications without modifying them. Ephemeral Addresses functions can contribute to various types of security enhancements that include privacy protections etc.

2. Definitions and Characteristics of Ephemeral Addresses

Definitions of Ephemeral Addresses are deprived from those of ephemeral ports. They are almost same as definitions of ephemeral ports. Only the differences are located in which layer they are used. Ephemeral ports are used as ports on the transport layer. On the other hand, Ephemeral Addresses are used as addresses on the network layer.

2.1 Where Ephemeral Addresses are used

Since ephemeral ports are used as clients’ source ports of TCP / UDP sessions on client nodes, Ephemeral Addresses are used as clients’ source addresses of TCP / UDP sessions on client nodes.

2.2 When Ephemeral Addresses are generated, assigned and disposed.

Since ephemeral ports are generated and assigned at when sessions are initiated on client nodes to communicate with server nodes, Ephemeral Addresses are generated and assigned at when sessions are initiated on client nodes to communicate with server nodes.

Since ephemeral ports are disposed on client nodes at when the sessions are closed, Ephemeral Addresses are also disposed on
client nodes at when the sessions are closed.

2.3 Effects to current applications and their programming styles

In typical client applications, source ports and addresses for their sessions are not specified.

When client applications do not specify source ports, the OS on the client node picks up and assigns appropriate source ports for their sessions automatically. (Such ports are called "ephemeral ports".)

If the kernel of the OS implemented Ephemeral Address functions and client applications do not specify source address (typical cases), the OS on the client node picks up and assigns appropriate source addresses for their sessions automatically. Such addresses are called "Ephemeral Addresses".

Important things in above described issued that: client applications do not specify source address for their session and there is no programming codes to specify source addresses.

It means that we can introduce Ephemeral Addresses features without modifying current existing enormous applications.

3. Comparison of Ephemeral Addresses and Temporary Addresses

In [RFC4941], "Temporary Addresses" are defined in order to enhance the privacy protection. Compared with Ephemeral Addresses, Temporary Addresses have the following similar functions.

1. The addresses are only used for client nodes addresses.
2. The addresses have lifetime, and theirs usable period is limited.
3. The addresses can enhance the privacy protection.

Therefore, we compare them in detail as follows.

3.1. Comparison from Abstract Function Viewpoints

[Temporary Address]:

A client uses a Temporary Address to access MULTIPLE services that are provided by multiple servers. The address is basically RE-USED when the client accesses a new service.

Timings when the address is created and abolished are not clearly defined. Therefore, in the worst case, the following situation may happen. When the lifetime of the Temporary Address expires and the
address becomes invalid, a session may be suddenly terminated even if the session is still active. Temporary Address includes the above potential problems.

[Ephemeral Address]:

A client uses an Ephemeral Address to access a SINGLE service. Of course, it is provided by a single server. The address is basically NOT RE-UESD for other sessions. Timings when the address is created and disposed are very clearly defined, because their definitions are derived from "ephemeral ports" specifications, and no problems are reported on "ephemeral ports" functions now. Thus, it never happens that the session is suddenly terminated when the lifetime of the Ephemeral Addresses expire.

Temporary Addresses are basically designed for long period lifetime usages. As a result, it is designed as a "RE-USE" type address. Since its design is NOT carried through a "one-time" policy, it has potential problems. On the other hand, the design of Ephemeral Address is carried through a "one-time" policy. An Ephemeral Address do not have the same types of problems that Temporary Addresses have.

Since the lifetime value of an Ephemeral Address becomes comparatively shorter than that of Temporary Address, it entails the following features. It is difficult for crackers to attack sessions or nodes that have such short lifetime addresses. Since this feature is good from a security viewpoint, it becomes additional benefit of Ephemeral Addresses.

3.2. Comparison from Address Creation Rule Viewpoints

Since Temporary Address is basically created by using simple random numbers, there is no relationship among series of created addresses. Thus, it is almost impossible to tell which Temporary Address comes from which node. With this specification, Anonymity is provided, but this becomes an unwelcome feature for administrators who would like to manage address information.

On the other hand, in the case of Ephemeral Address, it is necessary to include "port equivalent" info into the address. By putting some rules in a method including such "port equivalent" info, it becomes possible to have some relationships among series of created addresses. In other words, it becomes possible for administrators who know such including rules to manage Ephemeral Addresses (this feature is called Pseudonymity). The Ephemeral Address can provide not only an Anonymity feature but also a Pseudonymity feature.
We can also say that Ephemeral Addresses specification is superior to Temporary Address specification on this viewpoint.

4. Future Work

A definition which address values are used for Ephemeral Addresses is not clarified in this document. It will be clarified a future issued document. Ephemeral Addresses are categorized into a dynamically generated addresses type. When we use Ephemeral Addresses, we will meet the same type of problems that dynamically generated addresses have. We can not easy to omit or avoid the DAD operation time. It takes time to start using Ephemeral Addresses. Optimistic DAD [RFC4429] will not become the perfect solution to solve above described problems. In future documents, we will discuss the relationships between dynamically generated addresses and DAD operations and provide a kind of clear solution to meet this problem.

5. Security Considerations

Security Considerations of Temporary Addresses [RFC4941] can also be applied to Ephemeral Addresses. Since Ephemeral Addresses can provide Pseudonymity features, it becomes much easier to administer them than to administer Temporary Addresses.

6. IANA Considerations

Address space for Ephemral Addresses may be assigned by the IANA

Appendix A. Implementations

The Ephemral Address specification has been implemented under the following environments, and its basic functionaries have been verified

<table>
<thead>
<tr>
<th>OS: FreeBSD6.2R (32bit / 64bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU: i386 / amd64</td>
</tr>
</tbody>
</table>

Acknowledgement

A part of this work is supported by the program: SCOPE (Strategic Information and Communications R&D Promotion Programme) operated by Ministry of Internal Affairs and Communications of JAPAN.
Internet Draft     IPv6 Ephemeral Addresses

References

Normative References


Informative References


Authors’ Addresses

Hiroshi Kitamura
Common Platform Software Resarch Laboratories, NEC Corporation
(Igarashi Building 4F) 11-5, Shibaura 2-Chome,
Minato-Ku, Tokyo 108-8557, JAPAN
University of Electro-Communciations
5-1 Chofugaoka 1-Chome, Chofu-shi, Tokyo 182-8585, JAPAN
Phone: +81 3 5476 9795
Fax: +81 3 5476 1005
Email: kitamura@da.jp.nec.com

Shingo Ata
Graduate School of Engineering, Osaka City University
3-3-138, Sugimoto, Sumiyoshi-Ku, Osaka 558-8585, JAPAN
Phone: +81 6 6605 2191
Fax: +81 6 6605 2191
Email: ata@info.eng.osaka-cu.ac.jp

Masayuki Murata
Graduate School of Information Science and Technology, Osaka Univ.
1-5 Yamadaoka, Suita, Osaka 565-0871, JAPAN
Phone: +81 6 6879 4542
Fax: +81 6 6879 4544
Email: murata@ist.osaka-u.ac.jp
Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.