Workgroup: Network Working Group
Internet-Draft: draft-ietf-v6ops-ula-02
Published: 18 April 2023
Intended Status: Informational
Expires: 20 October 2023
Authors: N. Buraglio
Energy Sciences Network
C. Cummings
Energy Sciences Network
R. White
Juniper Networks

Unintended Operational Issues With ULA

Abstract

The behavior of ULA addressing as defined by [RFC6724] is preferred below legacy IPv4 addressing, thus rendering ULA IPv6 deployment functionally unusable in IPv4 / IPv6 dual-stacked environments. The lack of a consistent and supportable way to manipulate this behavior, across all platforms and at scale is counter to the operational behavior of GUA IPv6 addressing on nearly all modern operating systems that leverage a preference model based on [RFC6724] .

Status of This Memo

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

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

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

This Internet-Draft will expire on 20 October 2023.

Copyright Notice

Table of Contents

1. Introduction

In modern IPv4 / IPv6 dual-stacked environments, ULA addressing and GUA IPv6 addressing exhibit opposite behavior, which creates difficulties in deployments leveraging ULA addressing where there exist network elements that are unable to or do not support basic user conficuration of source address selection. This conflicting behavior carries planning, operational, and security implications for environments requiring ULA addressing with IPv4/IPv6 dual-stack and prioritization of IPv6 traffic by default, as is the behavior with IPv6 GUA addressing.

2. Defining Well Known Unintended Operational Issues With ULA

The [RFC6724] definition is incomplete for ULA precedence if a host is operating in a dual-stack environment. As written, [RFC6724] section 10.3 states: "The default policy table gives IPv6 addresses higher precedence than IPv4 addresses. This means that applications will use IPv6 in preference to IPv4 when the two are equally suitable. An administrator can change the policy table to prefer IPv4 addresses by giving the ::ffff:0.0.0.0/96 prefix a higher precedence". Expected behavior would be that locally preferred ULA address space would be preferred over legacy IPv4, however this is not the case where address selection is not configurable. This presents an acute issue with any environment using ULA addressing along side legacy IPv4, in that it is counter to the standard expectations for legacy IPv4 / IPv6 dual-stack behavior of preferring IPv6, as is performed with GUA addressing. Further, [RFC6724] Section 10.6 states that this is resolvable by adding a site-specific policy to cause ULAs within a site to be preferred

over global addresses. While theoretically possible, this presents significant issues on devices with inaccessable configuration files as detailed below.

3. Operational Implications

There are demonstrated and easily repeatible uses cases of ULA not being configurable and locally preferred over legacy IPv4 in widely deployed operating systems as well as many IoT and operational technology platforms that necessitate the immediate update to [RFC6724] to better reflect the original intent of the RFC. As with most adjustments to standards, and using [RFC6724] itself as a measurment, this update will likely take between 8-20 years to become common enough for relatively consistent behavior within operating systems. As a reference, as of the time of this writing, it has been 10 years since [RFC6724] has been published but we continue to see existing commercial and open source operating systems exhibiting [RFC3484] behavior. While it should be noted that [RFC6724] defines a solution that is functional academically, operationally the solution of adjusting the address preference selection table is both operating system dependent and unable to be signalled by any network mechanism such as within a router advertisement, DHCPv6 option, or the like. This lack of an intra-protocol or network-based ability to adjust address selection preference, along with the inability to adjust a notable number of operating systems either programmatically or manually renders operational scalability of such a mechanism functionally untenable. It is anticipated that any update of [RFC6724]would require an additional 8-20 years to be fully realized and properly implemented in a majority of network connected systems. In addition, in the current versions of Linux, the priority table (gai.conf) still makes reference to [RFC3484], further demonstrating the long timeframe to have updates reflected in a current, modern, widely deployed operating system. Examples of such out-of-date behavior can be found in printers, cameras, fixed devices, IoT sensors, and longer lifecycle equipment. It is especially important to note this behavior in the long lifecycle equipment that exists in industrial control and operational techology environments due to their very long mean time to replacement. The core issue is the stated interpretation from gai.conf that has the following default:

```
#scopev4  <mask> <value>
#    Add another rule to the RFC 6724 scope table for IPv4 addresses.
#    By default the scope IDs described in section 3.2 in RFC 6724 are
#    used.  Changing these defaults should hardly ever be necessary.
#    The defaults are equivalent to:
#
#scopev4 ::ffff:169.254.0.0/112  2
#scopev4 ::ffff:127.0.0.0/104    2
#scopev4 ::ffff:0.0.0.0/96       14
```

Figure 1

Notice that they are interpreting the legacy IPv4 address range as "scopev4" and the prefix ::ffff:0.0.0.0/96 which has a higher precedence (35) in [RFC6724] then the ULA prefix of fc00::/7 (3). This results in legacy IPv4 being preferred over IPv6 ULA.

The operational outcome is the move to dual-stack with ULA is inconsistent and imparts unnecessary difficulty for both troubleshooting and creating the baseline expected behavior which are both requirements for deployments. This results in operational and engineering teams not gaining IPv6 experience as limited traffic is actually using IPv6, and security baseline expectations are inconsistent at best and haphazard at worst.

In practice, [RFC6724] imposes several operational shortcomings preventing both consistent and desired behavior. If we define "desired behavior" as IPv6 preference over legacy IPv4 for address and protocol selection, then the resulting implemented behavior, based on [RFC6724] , will fall short of that intent due to the lack of a consistent manner for adjusting source address selection across platforms, and at scale. Based on the current verbiage, dual-stacked hosts configured with both a legacy IPv4 address and an IPv6 ULA address, the resulting behavior will manifest as a host choosing IPv4 over ULA IPv6. This behavior deviates from the current goal of a host with legacy IPv4 address and also with an IPv6 address preferring IPv6 over IPv4, regardless of the IPv6 address sourcing from ULA or GUA. Operationally and strategically, this manifests as an impediment to deployment of IPv6 for many non-service provider and mobile networks phasing in dual-stacked (both legacy IPv4 and IPv6) networking with the expectation of consistent behavior (i.e. prefer IPv6 before legacy IPv4).

Other operational considerations are the use of the policy table detailed in section 2.1 of [RFC6724] . While conceptually, the intent was for a configurable, longest-match table to be adjusted as-needed. In practice, inconsistency and lack of availibility to modify the prefix policy table remains difficult across platforms,

and in some cases is completely impossible, which in turn is the root cause of most of the issues surrounding ULA addressing and its use and support in production environments. Embedded, proprietary, closed source, and IoT devices are especially difficult to adjust and are, in many cases, incapable of any adjustment whatsoever. Large scale manipulation of the policy table also remains out of the realm of realistic support for small and medium scale operators due to lack of ability to manipulate all the hosts and systems, or a lack of tooling and access.

Below is an example of a gai.conf file from a modern Linux installation as of 03 April 2022:

```
# Configuration for getaddrinfo(3).
#
# So far only configuration for the destination address sorting is neede
# RFC 3484 governs the sorting.  But the RFC also says that system
# administrators should be able to overwrite the defaults.  This can be
# achieved here.
#
# All lines have an initial identifier specifying the option followed by
# up to two values.  Information specified in this file replaces the
# default information.  Complete absence of data of one kind causes the
# appropriate default information to be used.  The supported commands in
#
# reload  <yes|no>
#    If set to yes, each getaddrinfo(3) call will check whether this fil
#    changed and if necessary reload.  This option should not really be
#    used.  There are possible runtime problems.  The default is no.
#
# label   <mask>   <value>
#    Add another rule to the RFC 3484 label table.  See section 2.1 in
#    RFC 3484.  The default is:
#
#label ::1/128       0
#label ::/0          1
#label 2002::/16     2
#label ::/96         3
#label ::ffff:0:0/96 4
#label fec0::/10     5
#label fc00::/7      6
#label 2001:0::/32   7
#
#    This default differs from the tables given in RFC 3484 by handling
#    (now obsolete) site-local IPv6 addresses and Unique Local Addresses
#    The reason for this difference is that these addresses are never
#    NATed while IPv4 site-local addresses most probably are.  Given
#    the precedence of IPv6 over IPv4 (see below) on machines having onl
#    site-local IPv4 and IPv6 addresses a lookup for a global address wo
#    see the IPv6 be preferred.  The result is a long delay because the
#    site-local IPv6 addresses cannot be used while the IPv4 address is
#    (at least for the foreseeable future) NATed.  We also treat Teredo
#    tunnels special.
#
# precedence  <mask>   <value>
#    Add another rule to the RFC 3484 precedence table.  See section 2.1
#    and 10.3 in RFC 3484.  The default is:
#
#precedence  ::1/128       50
#precedence  ::/0          40
#precedence  2002::/16     30
#precedence ::/96          20
```

```
#precedence ::ffff:0:0/96  10
#
#    For sites which prefer IPv4 connections change the last line to
#
#precedence ::ffff:0:0/96  100

#
# scopev4  <mask>  <value>
#    Add another rule to the RFC 6724 scope table for IPv4 addresses.
#    By default the scope IDs described in section 3.2 in RFC 6724 are
#    used.  Changing these defaults should hardly ever be necessary.
#    The defaults are equivalent to:
#
#scopev4 ::ffff:169.254.0.0/112  2
#scopev4 ::ffff:127.0.0.0/104    2
#scopev4 ::ffff:0.0.0.0/96       14
```

Figure 2

Several assumptions are made here and are largely based on interpretations of [RFC6724] but are not operationally relevant in modern networks. As this file or an equivalent structure within a given operating system is referenced, it dictates the behavior of the getaddrinfo() or analogous process. More specifically, where getaddrinfo() or comparable API is used, the sorting behavior should take into account both the source address of the requesting host as well as the destination addresses returned and sort according to both source and destination addressing, i.e, when a ULA address is returned, the source address selection should return and use a ULA address if available. Similarly, if a GUA address is returned the source address selection should return a GUA source address if available.

Here are some example failure modes:

1. ULA per [RFC6724] is less preferred (the Precedence value is lower) than all legacy IPv4 (represented by ::ffff:0:0/96 in the aforementioned table).

2. Because of the lower Precedence value of fc00::/7, if a host has legacy IPv4 enabled, it will use legacy IPv4 before using ULA.

3. A dual-stacked client will source the traffic from the legacy IPv4 address, meaning it will require a corresponding legacy IPv4 destination address.

Per number 3, even a host choosing a destination with A and AAAA DNS records, the host in question will choose the A record to get an legacy IPv4 address for the destination, meaning ULA IPv6 is rendered completely unused. It is also notable that Happy Eyeballs ([RFC8305]) will not change the source address selection process on a host. Happy Eyeballs will only modify the destination sorting process.

As a direct result of the described failure modes, and in addition to the aforementioned operational implications, use of ULA is not a viable option for dual-stack \ networking transition planning, large scale network modeling, network lab environments or other modes of emulating a large scale networking that runs both IPv4 and IPv6 concurrently.

4. IANA Considerations

None at this time.

5. Security Considerations

Such unexpected behavior can result in operational outcomes which can result in serious security and compliance issues and could, in some cases, result in disabling of IPv6 to acheive compliance and consistency. .

6. Acknowledgements

The authors acknowledge the work of Brian Carpenter, David Farmer, Bob Hinden, Mark Andrews, Eduard Vasilenko, and Mark Smith for participation in the technical discussions leading to this finding and Michael Ackermann, Tom Coffeen, Kevin Myers, Jay Stewart, Paul Gear, and Ed Horley for providing further testing and operational input.

7. References

7.1. Normative References

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, DOI 10.17487/RFC3484, February 2003, <https://www.rfc-editor.org/info/rfc3484>.

[RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <https://www.rfc-editor.org/info/rfc6724>.

[RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <https://www.rfc-editor.org/info/rfc8305>.

7.2. Informative References

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., 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>.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <https://www.rfc-editor.org/info/rfc4193>.

[RFC6598] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space", BCP 153, RFC 6598, DOI 10.17487/RFC6598, April 2012, <https://www.rfc-editor.org/info/rfc6598>.

Authors' Addresses

Nick Buraglio
Energy Sciences Network

Email: buraglio@es.net

Chris Cummings
Energy Sciences Network

Email: chriscummings@es.net

Russ White
Juniper Networks

Email: russ@riw.us