Internet Engineering Task Force (IETF)
Request for Comments: 6052
Updates: 4291
Category: Standards Track
ISSN: 2070-1721

C. Bao
CERNET Center/Tsinghua University
C. Huitema
Microsoft Corporation
M. Bagnulo
UC3M
M. Boucadair
France Telecom
X. Li
CERNET Center/Tsinghua University

October 2010

IPv6 Addressing of IPv4/IPv6 Translators

Abstract

This document discusses the algorithmic translation of an IPv6 address to a corresponding IPv4 address, and vice versa, using only statically configured information. It defines a well-known prefix for use in algorithmic translations, while allowing organizations to also use network-specific prefixes when appropriate. Algorithmic translation is used in IPv4/IPv6 translators, as well as other types of proxies and gateways (e.g., for DNS) used in IPv4/IPv6 scenarios.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

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/rfc6052.

Copyright Notice

Table of Contents

1. Introduction

This document is part of a series of IPv4/IPv6 translation documents. A framework for IPv4/IPv6 translation is discussed in [v4v6-FRAMEWORK], including a taxonomy of scenarios that will be used in this document. Other documents specify the behavior of various types of translators and gateways, including mechanisms for translating between IP headers and other types of messages that include IP addresses. This document specifies how an individual IPv6 address is translated to a corresponding IPv4 address, and vice versa, in cases where an algorithmic mapping is used. While specific types of devices are used herein as examples, it is the responsibility of the specification of such devices to reference this document for algorithmic mapping of the addresses themselves.

Section 2 describes the prefixes and the format of "IPv4-embedded IPv6 addresses", i.e., IPv6 addresses in which 32 bits contain an IPv4 address. This format is common to both "IPv4-converted" and "IPv4-translatable" IPv6 addresses. This section also defines the algorithms for translating addresses, and the text representation of IPv4-embedded IPv6 addresses.

Section 3 discusses the choice of prefixes, the conditions in which they can be used, and the use of IPv4-embedded IPv6 addresses with stateless and stateful translation.

Section 4 provides a summary of the discussions behind two specific design decisions, the choice of a null suffix and the specific value of the selected prefix.

Section 5 discusses security concerns.

In some scenarios, a dual-stack host will unnecessarily send its traffic through an IPv6/IPv4 translator. This can be caused by the host’s default address selection algorithm [RFC3484], referrals, or other reasons. Optimizing these scenarios for dual-stack hosts is for future study.

1.1. Applicability Scope

This document is part of a series defining address translation services. We understand that the address format could also be used by other interconnection methods between IPv6 and IPv4, e.g., methods based on encapsulation. If encapsulation methods are developed by the IETF, we expect that their descriptions will document their specific use of IPv4-embedded IPv6 addresses.

1.2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.3. Terminology

This document makes use of the following terms:

Address translator: any entity that has to derive an IPv4 address from an IPv6 address or vice versa. This applies not only to devices that do IPv4/IPv6 packet translation, but also to other entities that manipulate addresses, such as name resolution proxies (e.g., DNS64 [DNS64]) and possibly other types of Application Layer Gateways (ALGs).

IPv4-converted IPv6 addresses: IPv6 addresses used to represent IPv4 nodes in an IPv6 network. They are a variant of IPv4-embedded IPv6 addresses and follow the format described in Section 2.2.

IPv4-embedded IPv6 addresses: IPv6 addresses in which 32 bits contain an IPv4 address. Their format is described in Section 2.2.

IPv4/IPv6 translator: an entity that translates IPv4 packets to IPv6 packets, and vice versa. It may do "stateless" translation, meaning that there is no per-flow state required, or "stateful" translation, meaning that per-flow state is created when the first packet in a flow is received.

IPv4-translatable IPv6 addresses: IPv6 addresses assigned to IPv6 nodes for use with stateless translation. They are a variant of IPv4-embedded IPv6 addresses and follow the format described in Section 2.2.

Network-Specific Prefix: an IPv6 prefix assigned by an organization for use in algorithmic mapping. Options for the Network-Specific Prefix are discussed in Sections 3.3 and 3.4.

Well-Known Prefix: the IPv6 prefix defined in this document for use in an algorithmic mapping.

2. IPv4-Embedded IPv6 Address Prefix and Format

2.1. Well-Known Prefix

This document reserves a "Well-Known Prefix" for use in an algorithmic mapping. The value of this IPv6 prefix is:

64:ff9b::/96

2.2. IPv4-Embedded IPv6 Address Format

IPv4-converted IPv6 addresses and IPv4-translatable IPv6 addresses follow the same format, described here as the IPv4-embedded IPv6 address Format. IPv4-embedded IPv6 addresses are composed of a variable-length prefix, the embedded IPv4 address, and a variable-length suffix, as presented in the following diagram, in which PL designates the prefix length:

```
    +--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |PL| 0-------------32--40--48--56--64--72--80--88--96--104---------|
    +--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |32|     prefix    |v4(32)         | u | suffix                    |
    +--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |40|     prefix        |v4(24)     | u |(8)| suffix                |
    +--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |48|     prefix            |v4(16) | u | (16)  | suffix            |
    +--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |56|     prefix                |(8)| u |  v4(24)   | suffix        |
    +--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |64|     prefix                    | u |   v4(32)      | suffix    |
    +--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |96|     prefix                                    |    v4(32)     |
    +--+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 1

In these addresses, the prefix shall be either the "Well-Known Prefix" or a "Network-Specific Prefix" unique to the organization deploying the address translators. The prefixes can only have one of the following lengths: 32, 40, 48, 56, 64, or 96. (The Well-Known Prefix is 96 bits long, and can only be used in the last form of the table.)

Various deployments justify different prefix lengths with Network-Specific Prefixes. The trade-off between different prefix lengths are discussed in Sections 3.3 and 3.4.

Bits 64 to 71 of the address are reserved for compatibility with the host identifier format defined in the IPv6 addressing architecture [RFC4291]. These bits MUST be set to zero. When using a /96 Network-Specific Prefix, the administrators MUST ensure that the bits 64 to 71 are set to zero. A simple way to achieve that is to construct the /96 Network-Specific Prefix by picking a /64 prefix, and then adding 4 octets set to zero.

The IPv4 address is encoded following the prefix, most significant bits first. Depending of the prefix length, the 4 octets of the address may be separated by the reserved octet "u", whose 8 bits MUST be set to zero. In particular:

- When the prefix is 32 bits long, the IPv4 address is encoded in positions 32 to 63.
- When the prefix is 40 bits long, 24 bits of the IPv4 address are encoded in positions 40 to 63, with the remaining 8 bits in position 72 to 79.
- When the prefix is 48 bits long, 16 bits of the IPv4 address are encoded in positions 48 to 63, with the remaining 16 bits in position 72 to 87.
- When the prefix is 56 bits long, 8 bits of the IPv4 address are encoded in positions 56 to 63, with the remaining 24 bits in position 72 to 95.
- When the prefix is 64 bits long, the IPv4 address is encoded in positions 72 to 103.
- When the prefix is 96 bits long, the IPv4 address is encoded in positions 96 to 127.

There are no remaining bits, and thus no suffix, if the prefix is 96 bits long. In the other cases, the remaining bits of the address constitute the suffix. These bits are reserved for future extensions and SHOULD be set to zero. Address translators who receive IPv4-embedded IPv6 addresses where these bits are not zero SHOULD ignore the bits’ value and proceed as if the bits’ value were zero. (Future extensions may specify a different behavior.)

2.3. Address Translation Algorithms

IPv4-embedded IPv6 addresses are composed according to the following algorithm:

- Concatenate the prefix, the 32 bits of the IPv4 address, and the suffix (if needed) to obtain a 128-bit address.
- If the prefix length is less than 96 bits, insert the null octet "u" at the appropriate position (bits 64 to 71), thus causing the least significant octet to be excluded, as documented in Figure 1.

The IPv4 addresses are extracted from the IPv4-embedded IPv6 addresses according to the following algorithm:

- If the prefix is 96 bits long, extract the last 32 bits of the IPv6 address;
- For the other prefix lengths, remove the "u" octet to obtain a 120-bit sequence (effectively shifting bits 72-127 to positions 64-119), then extract the 32 bits following the prefix.

2.4. Text Representation

IPv4-embedded IPv6 addresses will be represented in text in conformity with Section 2.2 of [RFC4291]. IPv4-embedded IPv6 addresses constructed using the Well-Known Prefix or a /96 Network-Specific Prefix may be represented using the alternative form presented in Section 2.2 of [RFC4291], with the embedded IPv4 address represented in dotted decimal notation. Examples of such representations are presented in Tables 1 and 2.

Network-Specific Prefix	IPv4 address	IPv4-embedded IPv6 address
2001:db8::/32	192.0.2.33	2001:db8:c000:221::
2001:db8:100::/40	192.0.2.33	2001:db8:1c0:2:21::
2001:db8:122::/48	192.0.2.33	2001:db8:122:c000:2:2100::
2001:db8:122:300::/56	192.0.2.33	2001:db8:122:3c0:0:221::
2001:db8:122:344::/64	192.0.2.33	2001:db8:122:344:c0:2:2100::
2001:db8:122:344::/96	192.0.2.33	2001:db8:122:344::192.0.2.33

Table 1: Text Representation of IPv4-Embedded IPv6 Addresses Using Network-Specific Prefixes

Well-Known Prefix	IPv4 address	IPv4-Embedded IPv6 address
64:ff9b::/96	192.0.2.33	64:ff9b::192.0.2.33

Table 2: Text Representation of IPv4-Embedded IPv6 Addresses Using the Well-Known Prefix

The Network-Specific Prefix examples in Table 1 are derived from the IPv6 prefix reserved for documentation in [RFC3849]. The IPv4 address 192.0.2.33 is part of the subnet 192.0.2.0/24 reserved for documentation in [RFC5735]. The representation of IPv6 addresses is compatible with [RFC5952].

3. Deployment Guidelines

3.1. Restrictions on the Use of the Well-Known Prefix

The Well-Known Prefix MUST NOT be used to represent non-global IPv4 addresses, such as those defined in [RFC1918] or listed in Section 3 of [RFC5735]. Address translators MUST NOT translate packets in which an address is composed of the Well-Known Prefix and a non-global IPv4 address; they MUST drop these packets.

The Well-Known Prefix SHOULD NOT be used to construct IPv4-translatable IPv6 addresses. The nodes served by IPv4-translatable IPv6 addresses should be able to receive global IPv6 traffic bound to their IPv4-translatable IPv6 address without incurring intermediate protocol translation. This is only possible if the specific prefix used to build the IPv4-translatable IPv6 addresses is advertised in inter-domain routing, but the advertisement of more specific prefixes derived from the Well-Known Prefix is not supported, as explained in Section 3.2. Network-Specific Prefixes SHOULD be used in these scenarios, as explained in Section 3.3.

The Well-Known Prefix MAY be used by organizations deploying translation services, as explained in Section 3.4.

3.2. Impact on Inter-Domain Routing

The Well-Known Prefix MAY appear in inter-domain routing tables, if service providers decide to provide IPv6-IPv4 interconnection services to peers. Advertisement of the Well-Known Prefix SHOULD be controlled either by upstream and/or downstream service providers according to inter-domain routing policies, e.g., through configuration of BGP [RFC4271]. Organizations that advertise the Well-Known Prefix in inter-domain routing MUST be able to provide IPv4/IPv6 translation service.

When the IPv4/IPv6 translation relies on the Well-Known Prefix, IPv4-embedded IPv6 prefixes longer than the Well-Known Prefix MUST NOT be advertised in BGP (especially External BGP) [RFC4271] because this leads to importing the IPv4 routing table into the IPv6 one and therefore introduces scalability issues to the global IPv6 routing table. Administrators of BGP nodes SHOULD configure filters that discard advertisements of embedded IPv6 prefixes longer than the Well-Known Prefix.

When the IPv4/IPv6 translation service relies on Network-Specific Prefixes, the IPv4-translatable IPv6 prefixes used in stateless translation MUST be advertised with proper aggregation to the IPv6 Internet. Similarly, if translators are configured with multiple Network-Specific Prefixes, these prefixes MUST be advertised to the IPv6 Internet with proper aggregation.

3.3. Choice of Prefix for Stateless Translation Deployments

Organizations may deploy translation services using stateless translation. In these deployments, internal IPv6 nodes are addressed using IPv4-translatable IPv6 addresses, which enable them to be accessed by IPv4 nodes. The addresses of these external IPv4 nodes are then represented in IPv4-converted IPv6 addresses.

Organizations deploying stateless IPv4/IPv6 translation SHOULD assign a Network-Specific Prefix to their IPv4/IPv6 translation service. IPv4-translatable and IPv4-converted IPv6 addresses MUST be constructed as specified in Section 2.2. IPv4-translatable IPv6 addresses MUST use the selected Network-Specific Prefix. Both IPv4-translatable IPv6 addresses and IPv4-converted IPv6 addresses SHOULD use the same prefix.

Using the same prefix ensures that IPv6 nodes internal to the organization will use the most efficient paths to reach the nodes served by IPv4-translatable IPv6 addresses. Specifically, if a node learns the IPv4 address of a target internal node without knowing that this target is in fact located behind the same translator that the node also uses, translation rules will ensure that the IPv6 address constructed with the Network-Specific Prefix is the same as the IPv4-translatable IPv6 address assigned to the target. Standard routing preference (i.e., "most specific match wins") will then ensure that the IPv6 packets are delivered directly, without requiring that translators receive the packets and then return them in the direction from which they came.

The intra-domain routing protocol must be able to deliver packets to the nodes served by IPv4-translatable IPv6 addresses. This may require routing on some or all of the embedded IPv4 address bits. Security considerations detailed in Section 5 require that routers check the validity of the IPv4-translatable IPv6 source addresses, using some form of reverse path check.

The management of stateless address translation can be illustrated with a small example:

> We will consider an IPv6 network with the prefix 2001:db8:122::/48. The network administrator has selected the Network-Specific Prefix 2001:db8:122:344::/64 for managing stateless IPv4/IPv6 translation. The IPv4-translatable address block for IPv4 subnet 192.0.2.0/24 is 2001:db8:122:344:c0:2::/96. In this network, the host A is assigned the IPv4-translatable IPv6 address 2001:db8:122:344:c0:2:2100::, which corresponds to the IPv4 address 192.0.2.33. Host A’s address is configured either manually or through DHCPv6.
>
> In this example, host A is not directly connected to the translator, but instead to a link managed by a router R. The router R is configured to forward to A the packets bound to 2001:db8:122:344:c0:2:2100::. To receive these packets, R will advertise reachability of the prefix 2001:db8:122:344:c0:2:2100::/104 in the intra-domain routing protocol -- or perhaps a shorter prefix if many hosts on link have IPv4-translatable IPv6 addresses derived from the same IPv4 subnet. If a packet bound to 192.0.2.33 reaches the translator, the destination address will be translated to 2001:db8:122:344:c0:2:2100::, and the packet will be routed towards R and then to A.
>
> Let’s suppose now that a host B of the same domain learns the IPv4 address of A, maybe through an application-specific referral. If B has translation-aware software, B can compose a destination address by combining the Network-Specific Prefix 2001:db8:122:344::/64 and the IPv4 address 192.0.2.33, resulting in the address 2001:db8:122:344:c0:2:2100::. The packet sent by B will be forwarded towards R, and then to A, avoiding protocol translation.

Forwarding, and reverse path checks, are more efficient when performed on the combination of the prefix and the IPv4 address. In theory, routers are able to route on prefixes of any length, but in practice there may be routers for which routing on prefixes larger than 64 bits is slower. However, routing efficiency is not the only consideration in the choice of a prefix length. Organizations also need to consider the availability of prefixes, and the potential impact of all-zero identifiers.

If a /32 prefix is used, all the routing bits are contained in the top 64 bits of the IPv6 address, leading to excellent routing properties. These prefixes may however be hard to obtain, and allocation of a /32 to a small set of IPv4-translatable IPv6 addresses may be seen as wasteful. In addition, the /32 prefix and a zero suffix lead to an all-zero interface identifier, which is an issue that we discuss in Section 4.1.

Intermediate prefix lengths such as /40, /48, or /56 appear as compromises. Only some of the IPv4 bits are part of the /64 prefixes. Reverse path checks, in particular, may have a limited efficiency. Reverse path checks limited to the most significant bits of the IPv4 address will reduce the possibility of spoofing external IPv4 addresses, but would allow IPv6 nodes to spoof internal IPv4-translatable IPv6 addresses.

We propose a compromise, based on using no more than 1/256th of an organization’s allocation of IPv6 addresses for the IPv4/IPv6 translation service. For example, if the organization is an Internet Service Provider with an allocated IPv6 prefix /32 or shorter, the ISP could dedicate a /40 prefix to the translation service. An end site with a /48 allocation could dedicate a /56 prefix to the translation service, or possibly a /96 prefix if all IPv4-translatable IPv6 addresses are located on the same link.

The recommended prefix length is also a function of the deployment scenario. The stateless translation can be used for Scenario 1, Scenario 2, Scenario 5, and Scenario 6 defined in [v4v6-FRAMEWORK]. For different scenarios, the prefix length recommendations are:

- For Scenario 1 (an IPv6 network to the IPv4 Internet) and Scenario 2 (the IPv4 Internet to an IPv6 network), an ISP holding a /32 allocation SHOULD use a /40 prefix, and a site holding a /48 allocation SHOULD use a /56 prefix.

- For Scenario 5 (an IPv6 network to an IPv4 network) and Scenario 6 (an IPv4 network to an IPv6 network), the deployment SHOULD use a /64 or a /96 prefix.

3.4. Choice of Prefix for Stateful Translation Deployments

Organizations may deploy translation services based on stateful translation technology. An organization may decide to use either a Network-Specific Prefix or the Well-Known Prefix for its stateful IPv4/IPv6 translation service.

When these services are used, IPv6 nodes are addressed through standard IPv6 addresses, while IPv4 nodes are represented by IPv4-converted IPv6 addresses, as specified in Section 2.2.

The stateful nature of the translation creates a potential stability issue when the organization deploys multiple translators. If several translators use the same prefix, there is a risk that packets belonging to the same connection may be routed to different translators as the internal routing state changes. This issue can be avoided either by assigning different prefixes to different translators or by ensuring that all translators using the same prefix coordinate their state.

Stateful translation can be used in scenarios defined in [v4v6-FRAMEWORK]. The Well-Known Prefix SHOULD be used in these scenarios, with two exceptions:

- In all scenarios, the translation MAY use a Network-Specific Prefix, if deemed appropriate for management reasons.
- The Well-Known Prefix MUST NOT be used for Scenario 3 (the IPv6 Internet to an IPv4 network), as this would lead to using the Well-Known Prefix with non-global IPv4 addresses. That means a Network-Specific Prefix (for example, a /96 prefix) MUST be used in that scenario.

4. Design Choices

The prefix that we have chosen reflects two design choices, the null suffix and the specific value of the Well-Known Prefix. We provide here a summary of the discussions leading to those two choices.

4.1. Choice of Suffix

The address format described in Section 2.2 recommends a zero suffix. Before making this recommendation, we considered different options: checksum neutrality, the encoding of a port range, and a value different than 0.

In the case of stateless translation, there would be no need for the translator to recompute a one’s complement checksum if both the IPv4-translatable and the IPv4-converted IPv6 addresses were constructed in a "checksum-neutral" manner, that is, if the IPv6 addresses would have the same one’s complement checksum as the embedded IPv4 address. In the case of stateful translation, checksum neutrality does not eliminate checksum computation during translation, as only one of the two addresses would be checksum neutral. We considered reserving 16 bits in the suffix to guarantee checksum neutrality, but declined because it would not help with stateful translation and because checksum neutrality can also be achieved by an appropriate choice of the Network-Specific Prefix, i.e., selecting a prefix whose one's complement checksum equals either 0 or 0xffff.

There have been proposals to complement stateless translation with a port-range feature. Instead of mapping an IPv4 address to exactly one IPv6 prefix, the options would allow several IPv6 nodes to share an IPv4 address, with each node managing a different range of ports. If a port range extension is needed, it could be defined later, using bits currently reserved as null in the suffix.

When a /32 prefix is used, an all-zero suffix results in an all-zero interface identifier. We understand the conflict with Section 2.6.1 of RFC4291, which specifies that all zeroes are used for the subnet-router anycast address. However, in our specification, there is only one node with an IPv4-translatable IPv6 address in the /64 subnet, so the anycast semantic does not create confusion. We thus decided to keep the null suffix for now. This issue does not exist for prefixes larger than 32 bits, such as the /40, /56, /64, and /96 prefixes that we recommend in Section 3.3.

4.2. Choice of the Well-Known Prefix

Before making our recommendation of the Well-Known Prefix, we were faced with three choices:

- reuse the IPv4-mapped prefix, ::ffff:0:0/96, as specified in RFC 2765, Section 2.1;
- request IANA to allocate a /32 prefix, or
- request allocation of a new /96 prefix.

We weighted the pros and cons of these choices before settling on the recommended /96 Well-Known Prefix.

The main advantage of the existing IPv4-mapped prefix is that it is already defined. Reusing that prefix would require minimal standardization efforts. However, being already defined is not just an advantage, as there may be side effects of current implementations. When presented with the IPv4-mapped prefix, current versions of Windows and Mac OS generate IPv4 packets, but will not send IPv6 packets. If we used the IPv4-mapped prefix, these nodes would not be able to support translation without modification. This will defeat the main purpose of the translation techniques. We thus eliminated the first choice, i.e., decided to not reuse the IPv4-mapped prefix, ::ffff:0:0/96.

A /32 prefix would have allowed the embedded IPv4 address to fit within the top 64 bits of the IPv6 address. This would have facilitated routing and load balancing when an organization deploys several translators. However, such destination-address-based load balancing may not be desirable. It is not compatible with Session Traversal Utilities for NAT (STUN) [RFC5389] in the deployments involving multiple stateful translators, each one having a different pool of IPv4 addresses. STUN compatibility would only be achieved if the translators managed the same pool of IPv4 addresses and were able to coordinate their translation state, in which case there is no big advantage to using a /32 prefix rather than a /96 prefix.

According to Section 2.2 of [RFC4291], in the legal textual representations of IPv6 addresses, dotted decimal can only appear at the end. The /96 prefix is compatible with that requirement. It enables the dotted decimal notation without requiring an update to [RFC4291]. This representation makes the address format easier to use and the log files easier to read.

The prefix that we recommend has the particularity of being "checksum neutral". The sum of the hexadecimal numbers "0064" and "ff9b" is "ffff", i.e., a value equal to zero in one’s complement arithmetic. An IPv4-embedded IPv6 address constructed with this prefix will have the same one’s complement checksum as the embedded IPv4 address.

5. Security Considerations

5.1. Protection against Spoofing

IPv4/IPv6 translators can be modeled as special routers, are subject to the same risks, and can implement the same mitigations. (The discussion of generic threats to routers and their mitigations is beyond the scope of this document.) There is, however, a particular risk that directly derives from the practice of embedding IPv4 addresses in IPv6: address spoofing.

An attacker could use an IPv4-embedded IPv6 address as the source address of malicious packets. After translation, the packets will appear as IPv4 packets from the specified source, and the attacker may be hard to track. If left without mitigation, the attack would allow malicious IPv6 nodes to spoof arbitrary IPv4 addresses.

The mitigation is to implement reverse path checks and to verify throughout the network that packets are coming from an authorized location.

5.2. Secure Configuration

The prefixes used for address translation are used by IPv6 nodes to send packets to IPv6/IPv4 translators. Attackers could attempt to fool nodes, DNS gateways, and IPv4/IPv6 translators into using wrong values for these parameters, resulting in network disruption, denial of service, and possible information disclosure. To mitigate such attacks, network administrators need to ensure that prefixes are configured in a secure way.

The mechanisms for achieving secure configuration of prefixes are beyond the scope of this document.

5.3. Firewall Configuration

Many firewalls and other security devices filter traffic based on IPv4 addresses. Attackers could attempt to fool these firewalls by sending IPv6 packets to or from IPv6 addresses that translate to the filtered IPv4 addresses. If the attack is successful, traffic that was previously blocked might be able to pass through the firewalls disguised as IPv6 packets. In all such scenarios, administrators should assure that packets that send to or from IPv4-embedded IPv6 addresses are subject to the same filtering as those directly sent to or from the embedded IPv4 addresses.

The mechanisms for configuring firewalls and security devices to achieve this filtering are beyond the scope of this document.

6. IANA Considerations

IANA has made the following changes in the "Internet Protocol Version 6 Address Space" registry located at http://www.iana.org.

OLD:

IPv6 Prefix	Allocation	Reference	Note
0000::/8	Reserved by IETF	[RFC4291]	[1][5]

NEW:

IPv6 Prefix	Allocation	Reference	Note
0000::/8	Reserved by IETF	[RFC4291]	[1][5][6]

[6] The "Well-Known Prefix" 64:ff9b::/96 used in an algorithmic mapping between IPv4 to IPv6 addresses is defined out of the 0000::/8 address block, per RFC 6052.

7. Acknowledgements

Many people in the BEHAVE WG have contributed to the discussion that led to this document, including Andrew Sullivan, Andrew Yourtchenko, Ari Keranen, Brian Carpenter, Charlie Kaufman, Dan Wing, Dave Thaler, David Harrington, Ed Jankiewicz, Fred Baker, Hiroshi Miyata, Iljitsch van Beijnum, John Schnizlein, Keith Moore, Kevin Yin, Magnus Westerlund, Margaret Wasserman, Masahito Endo, Phil Roberts, Philip Matthews, Remi Denis-Courmont, Remi Despres, and William Waites.

Marcelo Bagnulo is partly funded by Trilogy, a research project supported by the European Commission under its Seventh Framework Program.

8. Contributors

The following individuals co-authored documents from which text has been incorporated, and are listed in alphabetical order.

Dave Thaler
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
USA
Phone: +1 425 703 8835
EMail: dthaler@microsoft.com

Fred Baker
Cisco Systems
Santa Barbara, California 93117
USA
Phone: +1-408-526-4257
Fax: +1-413-473-2403
EMail: fred@cisco.com

Hiroshi Miyata
Yokogawa Electric Corporation
2-9-32 Nakacho
Musashino-shi, Tokyo 180-8750
JAPAN
EMail: h.miyata@jp.yokogawa.com

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

9.2. Informative References

[DNS64] Bagnulo, M., Sullivan, A., Matthews, P., and I. Beijnum, "DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", Work in Progress, October 2010.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

[RFC3849] Huston, G., Lord, A., and P. Smith, "IPv6 Address Prefix Reserved for Documentation", RFC 3849, July 2004.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[RFC5735] Cotton, M. and L. Vegoda, "Special Use IPv4 Addresses", BCP 153, RFC 5735, January 2010.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.

[v4v6-FRAMEWORK] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", Work in Progress, August 2010.

Authors' Addresses

Congxiao Bao
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing, 100084
China
Phone: +86 10-62785983
EMail: congxiao@cernet.edu.cn

Christian Huitema
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
U.S.A.
EMail: huitema@microsoft.com

Marcelo Bagnulo
UC3M
Av. Universidad 30
Leganes, Madrid 28911
Spain
Phone: +34-91-6249500
EMail: marcelo@it.uc3m.es
URI: http://www.it.uc3m.es/marcelo

Mohamed Boucadair
France Telecom
3, Av Francois Chateaux
Rennes 350000
France
EMail: mohamed.boucadair@orange-ftgroup.com

Xing Li
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing, 100084
China
Phone: +86 10-62785983
EMail: xing@cernet.edu.cn