Behave WG T. Savolainen
Internet-Draft Nokia
Intended status: Standards Track J. Korhonen
Expires: April 16, 2012 Nokia Siemens Networks
October 14, 2011

Discovery of a Network-Specific NAT64 Prefix using a Well-Known Name
draft-ietf-behave-nat64-discovery-heuristic-03.txt

Abstract

This document describes a method for detecting presence of DNS64 and for learning IPv6 prefix used for protocol translation on an access network without explicit support from the access network. The method depends on existence of a well-known IPv4-only domain name. The information learned enables applications and hosts to perform local IPv6 address synthesis and on dual-stack accesses avoid traversal through NAT64.

Table of Contents

1. Introduction 3
2. Requirements and Terminology 3
2.1. Requirements 3
2.2. Terminology 4
3. Host behavior 4
3.1. Connectivity test 5
4. Operational considerations for hosting the IPv4-only well-known name 6
5. DNS(64) entity considerations 6
6. Exit strategy 6
7. Security Considerations 6
8. IANA Considerations 7
8.1. About the IPv4 address for the well-known name 7
9. Acknowledgements 7
10. Normative References 8
Authors' Addresses 8

1. Introduction

As part of the transition to IPv6 NAT64 [RFC6146] and DNS64 [RFC6147] technologies will be utilized by some access networks to provide IPv4 connectivity for IPv6-only hosts. The DNS64 utilizes IPv6 address synthesis to create local IPv6 presentations of peers having only IPv4 addresses, hence allowing DNS-using IPv6-only hosts to communicate with IPv4-only peers.

However, DNS64 cannot serve applications not using DNS, such as those receiving IPv4 address literals as referrals. Such applications could nevertheless be able to work through NAT64, provided they are able to create locally valid IPv6 presentations of peers' IPv4 addresses.

Additionally, DNS64 is not able to do IPv6 address synthesis for hosts running validating DNSSEC enabled resolvers, but instead the synthesis must be done by the hosts themselves. In order to perform IPv6 synthesis hosts have to learn the IPv6 prefix(es) used on the access network for protocol translation.

This document describes a best effort method for applications and hosts to learn the information required to perform local IPv6 address synthesis. An example application is a browser encountering IPv4 address literals in an IPv6-only access network. Another example is a host running validating security aware DNS resolver in an IPv6-only access network.

The knowledge of IPv6 address synthesis taking place may also be useful if DNS64 and NAT64 are present in dual-stack enabled access networks. In such cases hosts may choose to prefer IPv4 in order to avoid traversal through protocol translators.

It is important to notice that use of this approach will not result in as robust and good behaving system as an all-IPv6 system would be. Hence it is highly RECOMMENDED to upgrade to IPv6 and utilize the described method only as a short-term solution.

2. Requirements and Terminology

2.1. Requirements

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 [RFC2119].

2.2. Terminology

Well-Known IPv4-only Name (WKN): a fully qualified domain name well-known to have only A record.

Well-Known IPv4 Address: an IPv4 address that is well-known and mapped to the well-known name.

3. Host behavior

A host requiring information about presence of NAT64 and the IPv6 prefix used for protocol translation shall send a DNS query for AAAA records of a well-known IPv4-only fully qualified domain name. This may happen, for example, at the moment the host is configured an IPv6 address of a DNS server. This may also happen at the time when first DNS query for AAAA record is initiated. The host may perform this check in both IPv6-only and dual-stack access networks.

When sending AAAA query for the known name a host MUST set "Checking Disabled (CD)" bit to zero, as otherwise the DNS64 will not perform IPv6 address synthesis hence does not reveal the IPv6 prefix(es) used for protocol translation.

A DNS reply with one or more non-empty AAAA records indicates that the access network is utilizing IPv6 address synthesis. A host MUST look through all of the received AAAA records to collect all available prefixes. The prefixes may include Well-Known Prefix 64:ff9b::/96 [RFC6052] or one or more Network-Specific Prefixes. In the case of NSPs the host SHALL search for the IPv4 address inside of the received IPv6 addresses to determine used address format.

An IPv4 address inside synthesized IPv6 address should be found at some of the locations described in [RFC6052]. If the searched IPv4 address is not found on any of the standard locations the network must be using different formatting. Developers may over time learn on IPv6 translated address formats that are extensions or alternatives to the standard formats. Developers MAY at that point add additional steps to the described discovery procedures. The additional steps are outside the scope of the present document.

The host should ensure a 32-bit IPv4 address value is present only once in an IPv6 address. In case another instance of the value is found inside the IPv6, the host shall repeat the search with another IPv4 address, if possible.

In the case only one IPv6 prefix was present in the DNS response: a host shall use that IPv6 prefix for both local synthesis and for

detecting synthesis done by the DNS64 entity on the network.

In the case multiple IPv6 prefixes were present in the DNS response: a host SHOULD use all received prefixes when determining whether other received IPv6 addresses are synthetic. However, for selecting prefix for the local IPv6 address synthesis host MUST use the following prioritization order, of which purpose is to avoid use of prefixes containing suffixes reserved for the future [RFC6052]:

1. Use NSP having /96 prefix

2. Use WKP prefix

3. Use longest available NSP prefix

In the case of NXDOMAIN response or an empty AAAA reply: the DNS64 is not available on the access network, network filtered the well-known query on purpose, or something went wrong in the DNS resolution. All unsuccessful cases result in unavailability of a host to perform local IPv6 address synthesis. The host MAY periodically resend AAAA query to check if DNS64 has become available or possibly temporary problem cleared. The host MAY perform A query for the well-known name to learn whether the service is available at all (see section 6 about Exit Strategy). The host MAY also continue monitoring DNS replies with IPv6 addresses constructed from WKP, in which case the host MAY use the WKP as if it were learned during the query for well-known name.

To save Internet's resources, if possible, a host should perform NAT64 discovery only when needed (e.g. when local synthesis is required, cached reply timeouts, new network interface is started, and so forth. Furthermore, the host SHOULD cache the replies it receives and honor TTLs.

3.1. Connectivity test

After the host has obtained a candidate prefix and format for the IPv6 address synthesis it may locally synthesize an IPv6 address, by using a publicly routable IPv4 address, and test connectivity with the resulting IPv6 address. The connectivity test may be conducted e.g. with ICMPv6 or with a transport layer protocol.

This connectivity test ensures local address synthesis results in functional and protocol translatable IPv6 addresses.

The host MUST NOT perform connectivity test for the well-known IPv4 address of the well-known name, but instead to some other destination such as host vendor servers.

In many scenarios separate connectivity test is not really required as an application may just try to connect to the IPv4-only destination with synthetic IPv6 address and see if a connection is successfully established or not.

4. Operational considerations for hosting the IPv4-only well-known name

The authoritative name server for the well-known name shall have DNS record TTL set to a long value in order to improve effectiveness of DNS caching and robustness of the discovery procedure in general. The exact value depends on availability time for the used public IPv4 address, but should not be longer than one year.

The domain serving the well-known name must be signed with DNSSEC. See also Security Considerations section.

It is expected that volumes for well-known name related queries are roughly SOMETHING, TBD. The infrastructure required to serve well-known name is SOMETHING, TBD.

5. DNS(64) entity considerations

DNS(64) servers MUST NOT interfere or perform special procedures for the queries related to the well-known name until the time has arrived for the exit strategy to be deployed.

6. Exit strategy

A day will come when this tool is no longer needed. At that point best suited techniques for implementing exit strategy will be documented. In the global scope the exit strategy may include sending NXDOMAIN replies by the authoritative name server of the well-known name with a very long TTL.

A client implementation receiving NXDOMAIN response for the A query of the well-known name means SHOULD consider this tool as disabled.

7. Security Considerations

The security considerations follow closely those of RFC6147 [RFC6147]. If an attacker manages to change the NSP prefix host discovers, the traffic generated by the host will be delivered to altered destination. This can result in either a denial-of-service (DoS) attack (if the resulting IPv6 addresses are not assigned to any

device), a flooding attack (if the resulting IPv6 addresses are assigned to devices that do not wish to receive the traffic), or an eavesdropping attack (in case the altered NSP is routed through the attacker).

The zone serving the well-known name has to be protected with DNSSEC, as otherwise it will be too attractive target for attackers who wish to alter hosts' NSP prefix discovery procedures.

A host SHOULD implement validating DNSSEC resolver for validating the A response of the well-known name query. A host without validating DNSSEC resolver SHOULD request validation to be performed by the used recursive DNS server.

8. IANA Considerations

A well-known name should be defined and a public IPv4 address allocated (by IANA? IETF? Someone else?).

8.1. About the IPv4 address for the well-known name

The global IPv4 address for the well-known, if possible, should be chosen so that it is unlikely to appear more than once within an IPv6 address and also as easy as possible to find from within the synthetic IPv6 address. A global address is required as otherwise DNS64 entity will not perform AAAA record synthesis. The address does not have to be routable as no communications are initiated to the IPv4 address.

Allocating two IPv4 addresses would improve the heuristics in cases where the primary IPv4 address' bit pattern appears more than once in the synthetic IPv6 address (NSP prefix contains the same bit pattern as the IPv4 address).

If no well-known IPv4 address is allocated for this method, the heuristic requires sending additional A query to learn the IPv4 address that is sought inside the received IPv6 address. Without knowing IPv4 address it is impossible to determine address format used by DNS64.

9. Acknowledgements

Authors would like to thank Andrew Sullivan, Dan Wing, Washam Fan, Cameron Byrne, Zhenqiang Li, Dave Thaler, and Christian Huitema for significant improvement ideas and comments.

10. Normative References

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

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.

Authors' Addresses

Teemu Savolainen
Nokia
Hermiankatu 12 D
FI-33720 Tampere
Finland

Email: teemu.savolainen@nokia.com

Jouni Korhonen
Nokia Siemens Networks
Linnoitustie 6
FI-02600 Espoo
Finland

Email: jouni.nospam@gmail.com