Internet Engineering Task Force
INTERNET-DRAFT
Feb 21, 2003
Expires Aug 2, 2003

Alain Durand
SUN Microsystems

Dynamic reverse DNS for IPv6

<draft-durand-dnsop-dynreverse-00.txt>

Status of this memo

Abstract

This document describes a method to dynamically generate PTR records and corresponding A or AAAA records when the reverse path DNS tree is not populated.

A special domain dynrev.arpa. is reserved for that purpose.

1. Introduction

In IPv4, the reverse path tree of the DNS under in-addr.arpa. although not perfectly maintained, is still mostly usable and its existence is important for a number of applications that relies on its existence and decent status. Some applications performs some (very) weak security checks based on it. Mail relays relies on it for some anti-spams checks an some FTP server will not let you in unless your IP address resolve properly with a PTR record.

IPv6 addresses being much longer (and cumbersome) than IPv4 addresses, it is to fear that the reverse path tree under ip6.arpa. would not be as well maintained. Also, tools like 6to4, Isatap and others have made creative use of the 128 bits of an IPv6 address to automatically embed an IPv4 address to enable seamless connection to the IPv6 Internet. However, no provision has been made to make sure the reverse path tree gets automatically updated as well for those

new IPv6 addresses. One step furter, RFC3041 describes a mechanism to basically use random bits in the bottom part of an IPv6 address to preserver anonymity. If those addresses are to resolve in the reverse path tree, it obviously has to be with anonymous data as well. Another point to note is that home customer ISPs in IPv4 have a current practice to pre-populate the reverse path tree with names automatically derived from the IP addresses. This practice is no longer possible in IPv6, where IP address allocation is not dense as it is the case in IPv4. The mere size of typical customer allocation (2^48 according to the recommendation of RFC3177) makes it impossible.

Applications that check the existence of PTR records usually follow this by checking if the name pointed by the PTR resolve in a A (or AAAA for IPv6) that match the original IP address. Thus the forward path tree must also include the corresponding data.

One simple approach of this problem is to simply declare the usage of the reverse path DNS as described above obsolete. The author believe this is too strong an approach for now.

Similarly, a completely different approach would be to deprecate the usage of DNS for the reverse tree altogether and replace it by something inspired from ICMP name-info messages. The author believes that this approached is an important departure from the current practise and thus not very realistic. Also, there are some concerns about the the security implications of this method as any node could easily impersonate any name. This approach would fundamentally change the underlying assumption of "I trust what has been put in the DNS by the local administrators" to "I trust what has been configured on each machine I query directly".

2. Dynamic record generation

If static pre-population of the tree is not possible anymore and data still need to be returned to applications using getnameinfo(), the alternative is dynamic record generation. This can be done is two places: in the DNS servers responsible for the allocated space (/64 or /48) in the ip6.arpa. domain. or in the DNS resolvers (either the sub resolver library or the recursive DNS server).

2.1. On the resolver side.

The resolver, either in the recursive DNS server or in the stub library could theoretically generate this data.

In case DNSsec is in place, the recursive DNS server would have to pretend these records are authentic.

If the synthesis is done in the stub-resolver library, no record

needs to be actually generated, only the right information needs to be passed to getnameinfo() and getaddrinfo(). If the synthesis is done in the recursive DNS server, no modification is required to existing stub resolvers.

2.2. On the server side.

PTR records could be generated automatically by the server responsible for the reverse path tree of an IPv6 prefix (a /64 or /48 prefixes or basically anything in between) when static data is not available.

There could be impact on DNSsec as the zone or some parts of the zone may need to be resigned each time a DNS query is made for an unpopulated address. This can be seen as a DOS attack on a DNSsec zone, so server side synthesis is not recommended if DNSsec is deployed.

3. Synthesis

The algorithm is simple: Do the normal queries. If the query returns No such domain, replace this answer by the synthetized one if possible.

3.1. PTR synthesis

The synthetized PTR for a DNS string [X] is simply [X].dynrev.arpa. where [X] is any valid DNS name.

The fact that the synthetized PTR points to the dynrev.arpa. domain is an indication to the applications that this record has been dynamically generated.

3.2. A synthesis

If [X] is in the form a.b.c.d.in-addr.arpa, one can synthetized an A record for the string [X].dynrev.arpa. which value is d.c.b.a. with a,b,c & d being integer [0..255]

3.3. AAAA synthesis

If [X] is in the form a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.q.s.t.u.v.w.x.y.z.A.B.C.D.E.F.in-addr.arpa, one can synthetized a AAAA record for the string [X].dynrev.arpa. which value is FEDC:BAzy:xwvu:tsrq:ponm:lkji:hgfe:dcba with a,b,c....x,y,z,A,B,C,D,E,F being hexadecimal digits.

3.4. Server side synthesis

If synthesis is done on the server side, PTR could be set not to use the dynrev.arpa domain but the local domain name instead. It culd be for instance dynrev.mydomain.com.

Note also that server side synthesis is not incompatible with resolver side synthesis.

4. IANA considerations

The dynrev.arpa. domain is reserved for the purpose of this document.

5. Security considerations

Section 2. discusses the the interactions with DNSsec.

6. Authors addresses

Alain Durand
SUN Microsystems, Inc
17, Network Circle
UMPK17-202
Menlo Park, CA 94025
USA
Mail: Alain.Durand@sun.com