                    Non-Terminal DNS Name Redirection

Status of this Memo

Copyright Notice

1. Introduction

   This document defines a new DNS Resource Record called "DNAME", which
   provides the capability to map an entire subtree of the DNS name
   space to another domain. It differs from the CNAME record which maps
   a single node of the name space.

   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 [KWORD].

2. Motivation

   This Resource Record and its processing rules were conceived as a
   solution to the problem of maintaining address-to-name mappings in a
   context of network renumbering. Without the DNAME mechanism, an
   authoritative DNS server for the address-to-name mappings of some
   network must be reconfigured when that network is renumbered. With
   DNAME, the zone can be constructed so that it needs no modification
   when renumbered. DNAME can also be useful in other situations, such
   as when an organizational unit is renamed.

3. The DNAME Resource Record

   The DNAME RR has mnemonic DNAME and type code 39 (decimal).

   DNAME has the following format:

      <owner> <ttl> <class> DNAME <target>

   The format is not class-sensitive. All fields are required. The
   RDATA field <target> is a <domain-name> [DNSIS].

   The DNAME RR causes type NS additional section processing.

   The effect of the DNAME record is the substitution of the record's
   <target> for its <owner> as a suffix of a domain name. A "no-
   descendants" limitation governs the use of DNAMEs in a zone file:

      If a DNAME RR is present at a node N, there may be other data at N
      (except a CNAME or another DNAME), but there MUST be no data at
      any descendant of N. This restriction applies only to records of
      the same class as the DNAME record.

   This rule assures predictable results when a DNAME record is cached
   by a server which is not authoritative for the record's zone. It
   MUST be enforced when authoritative zone data is loaded. Together
   with the rules for DNS zone authority [DNSCLR] it implies that DNAME
   and NS records can only coexist at the top of a zone which has only
   one node.

   The compression scheme of [DNSIS] MUST NOT be applied to the RDATA
   portion of a DNAME record unless the sending server has some way of
   knowing that the receiver understands the DNAME record format.
   Signalling such understanding is expected to be the subject of future
   DNS Extensions.

   Naming loops can be created with DNAME records or a combination of
   DNAME and CNAME records, just as they can with CNAME records alone.
   Resolvers, including resolvers embedded in DNS servers, MUST limit
   the resources they devote to any query. Implementors should note,
   however, that fairly lengthy chains of DNAME records may be valid.

4. Query Processing

   To exploit the DNAME mechanism the name resolution algorithms [DNSCF]
   must be modified slightly for both servers and resolvers.

   Both modified algorithms incorporate the operation of making a
   substitution on a name (either QNAME or SNAME) under control of a
   DNAME record. This operation will be referred to as "the DNAME
   substitution".

4.1. Processing by Servers

   For a server performing non-recursive service steps 3.c and 4 of
   section 4.3.2 [DNSCF] are changed to check for a DNAME record before
   checking for a wildcard ("*") label, and to return certain DNAME
   records from zone data and the cache.

   DNS clients sending Extended DNS [EDNS0] queries with Version 0 or
   non-extended queries are presumed not to understand the semantics of
   the DNAME record, so a server which implements this specification,
   when answering a non-extended query, SHOULD synthesize a CNAME record
   for each DNAME record encountered during query processing to help the
   client reach the correct DNS data. The behavior of clients and
   servers under Extended DNS versions greater than 0 will be specified
   when those versions are defined.

   The synthesized CNAME RR, if provided, MUST have

      The same CLASS as the QCLASS of the query,

      TTL equal to zero,

      An <owner> equal to the QNAME in effect at the moment the DNAME RR
      was encountered, and

      An RDATA field containing the new QNAME formed by the action of
      the DNAME substitution.

   If the server has the appropriate key on-line [DNSSEC, SECDYN], it
   MAY generate and return a SIG RR for the synthesized CNAME RR.

   The revised server algorithm is:

   1. Set or clear the value of recursion available in the response
      depending on whether the name server is willing to provide
      recursive service. If recursive service is available and
      requested via the RD bit in the query, go to step 5, otherwise
      step 2.

   2. Search the available zones for the zone which is the nearest
      ancestor to QNAME. If such a zone is found, go to step 3,
      otherwise step 4.

   3. Start matching down, label by label, in the zone. The matching
      process can terminate several ways:

a. If the whole of QNAME is matched, we have found the node.

   If the data at the node is a CNAME, and QTYPE doesn't match
   CNAME, copy the CNAME RR into the answer section of the
   response, change QNAME to the canonical name in the CNAME RR,
   and go back to step 1.

   Otherwise, copy all RRs which match QTYPE into the answer
   section and go to step 6.

b. If a match would take us out of the authoritative data, we have
   a referral. This happens when we encounter a node with NS RRs
   marking cuts along the bottom of a zone.

   Copy the NS RRs for the subzone into the authority section of
   the reply. Put whatever addresses are available into the
   additional section, using glue RRs if the addresses are not
   available from authoritative data or the cache. Go to step 4.

c. If at some label, a match is impossible (i.e., the
   corresponding label does not exist), look to see whether the
   last label matched has a DNAME record.

   If a DNAME record exists at that point, copy that record into
   the answer section. If substitution of its <target> for its
   <owner> in QNAME would overflow the legal size for a <domain-
   name>, set RCODE to YXDOMAIN [DNSUPD] and exit; otherwise
   perform the substitution and continue. If the query was not
   extended [EDNS0] with a Version indicating understanding of the
   DNAME record, the server SHOULD synthesize a CNAME record as
   described above and include it in the answer section. Go back
   to step 1.

   If there was no DNAME record, look to see if the "*" label
   exists.

   If the "*" label does not exist, check whether the name we are
   looking for is the original QNAME in the query or a name we
   have followed due to a CNAME. If the name is original, set an
   authoritative name error in the response and exit. Otherwise
   just exit.

   If the "*" label does exist, match RRs at that node against
   QTYPE. If any match, copy them into the answer section, but
   set the owner of the RR to be QNAME, and not the node with the
   "*" label. Go to step 6.

   4. Start matching down in the cache. If QNAME is found in the cache,
      copy all RRs attached to it that match QTYPE into the answer
      section. If QNAME is not found in the cache but a DNAME record is
      present at an ancestor of QNAME, copy that DNAME record into the
      answer section. If there was no delegation from authoritative
      data, look for the best one from the cache, and put it in the
      authority section. Go to step 6.

   5. Use the local resolver or a copy of its algorithm (see resolver
      section of this memo) to answer the query. Store the results,
      including any intermediate CNAMEs and DNAMEs, in the answer
      section of the response.

   6. Using local data only, attempt to add other RRs which may be
      useful to the additional section of the query. Exit.

   Note that there will be at most one ancestor with a DNAME as
   described in step 4 unless some zone's data is in violation of the
   no-descendants limitation in section 3. An implementation might take
   advantage of this limitation by stopping the search of step 3c or
   step 4 when a DNAME record is encountered.

4.2. Processing by Resolvers

   A resolver or a server providing recursive service must be modified
   to treat a DNAME as somewhat analogous to a CNAME. The resolver
   algorithm of [DNSCF] section 5.3.3 is modified to renumber step 4.d
   as 4.e and insert a new 4.d. The complete algorithm becomes:

   1. See if the answer is in local information, and if so return it to
      the client.

   2. Find the best servers to ask.

   3. Send them queries until one returns a response.

   4. Analyze the response, either:

      a. if the response answers the question or contains a name error,
         cache the data as well as returning it back to the client.

      b. if the response contains a better delegation to other servers,
         cache the delegation information, and go to step 2.

      c. if the response shows a CNAME and that is not the answer
         itself, cache the CNAME, change the SNAME to the canonical name
         in the CNAME RR and go to step 1.

    d. if the response shows a DNAME and that is not the answer
       itself, cache the DNAME. If substitution of the DNAME's
       <target> for its <owner> in the SNAME would overflow the legal
       size for a <domain-name>, return an implementation-dependent
       error to the application; otherwise perform the substitution
       and go to step 1.

    e. if the response shows a server failure or other bizarre
       contents, delete the server from the SLIST and go back to step
       3.

   A resolver or recursive server which understands DNAME records but
   sends non-extended queries MUST augment step 4.c by deleting from the
   reply any CNAME records which have an <owner> which is a subdomain of
   the <owner> of any DNAME record in the response.

5. Examples of Use

5.1. Organizational Renaming

   If an organization with domain name FROBOZZ.EXAMPLE became part of an
   organization with domain name ACME.EXAMPLE, it might ease transition
   by placing information such as this in its old zone.

        frobozz.example. DNAME   frobozz-division.acme.example.
                          MX   10   mailhub.acme.example.

   The response to an extended recursive query for www.frobozz.example
   would contain, in the answer section, the DNAME record shown above
   and the relevant RRs for www.frobozz-division.acme.example.

5.2. Classless Delegation of Shorter Prefixes

   The classless scheme for in-addr.arpa delegation [INADDR] can be
   extended to prefixes shorter than 24 bits by use of the DNAME record.
   For example, the prefix 192.0.8.0/22 can be delegated by the
   following records.

        $ORIGIN 0.192.in-addr.arpa.
        8/22   NS   ns.slash-22-holder.example.
        8   DNAME   8.8/22
        9   DNAME   9.8/22
        10   DNAME   10.8/22
        11   DNAME   11.8/22

   A typical entry in the resulting reverse zone for some host with
   address 192.0.9.33 might be

        $ORIGIN 8/22.0.192.in-addr.arpa.
        33.9   PTR   somehost.slash-22-holder.example.

   The same advisory remarks concerning the choice of the "/" character
   apply here as in [INADDR].

5.3. Network Renumbering Support

   If IPv4 network renumbering were common, maintenance of address space
   delegation could be simplified by using DNAME records instead of NS
   records to delegate.

        $ORIGIN new-style.in-addr.arpa.
        189.190   DNAME   in-addr.example.net.

        $ORIGIN in-addr.example.net.
        188   DNAME   in-addr.customer.example.

        $ORIGIN in-addr.customer.example.
        1   PTR   www.customer.example.
        2   PTR   mailhub.customer.example.
        ; etc ...

   This would allow the address space 190.189.0.0/16 assigned to the ISP
   "example.net" to be changed without the necessity of altering the
   zone files describing the use of that space by the ISP and its
   customers.

   Renumbering IPv4 networks is currently so arduous a task that
   updating the DNS is only a small part of the labor, so this scheme
   may have a low value. But it is hoped that in IPv6 the renumbering
   task will be quite different and the DNAME mechanism may play a
   useful part.

6. IANA Considerations

   This document defines a new DNS Resource Record type with the
   mnemonic DNAME and type code 39 (decimal). The naming/numbering
   space is defined in [DNSIS]. This name and number have already been
   registered with the IANA.

7. Security Considerations

   The DNAME record is similar to the CNAME record with regard to the
   consequences of insertion of a spoofed record into a DNS server or
   resolver, differing in that the DNAME's effect covers a whole subtree
   of the name space. The facilities of [DNSSEC] are available to
   authenticate this record type.

8. References

   [DNSCF]   Mockapetris, P., "Domain names - concepts and facilities",
             STD 13, RFC 1034, November 1987.

   [DNSCLR] Elz, R. and R. Bush, "Clarifications to the DNS
             Specification", RFC 2181, July 1997.

   [DNSIS]   Mockapetris, P., "Domain names - implementation and
             specification", STD 13, RFC 1035, November 1987.

   [DNSSEC] Eastlake, 3rd, D. and C. Kaufman, "Domain Name System
             Security Extensions", RFC 2065, January 1997.

   [DNSUPD] Vixie, P., Ed., Thomson, S., Rekhter, Y. and J. Bound,
             "Dynamic Updates in the Domain Name System", RFC 2136, April
             1997.

   [EDNS0]   Vixie, P., "Extensions mechanisms for DNS (EDNS0)", RFC
             2671, August 1999.

   [INADDR] Eidnes, H., de Groot, G. and P. Vixie, "Classless IN-
             ADDR.ARPA delegation", RFC 2317, March 1998.

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

   [SECDYN] D. Eastlake, 3rd, "Secure Domain Name System Dynamic
             Update", RFC 2137, April 1997.

9. Author's Address

   Matt Crawford
   Fermilab MS 368
   PO Box 500
   Batavia, IL 60510
   USA

   Phone: +1 630 840-3461
   EMail: crawdad@fnal.gov

10. Full Copyright Statement

   Copyright (C) The Internet Society (1999). All Rights Reserved.

   This document and translations of it may be copied and furnished to
   others, and derivative works that comment on or otherwise explain it
   or assist in its implementation may be prepared, copied, published
   and distributed, in whole or in part, without restriction of any
   kind, provided that the above copyright notice and this paragraph are
   included on all such copies and derivative works. However, this
   document itself may not be modified in any way, such as by removing
   the copyright notice or references to the Internet Society or other
   Internet organizations, except as needed for the purpose of
   developing Internet standards in which case the procedures for
   copyrights defined in the Internet Standards process must be
   followed, or as required to translate it into languages other than
   English.

   The limited permissions granted above are perpetual and will not be
   revoked by the Internet Society or its successors or assigns.

   This document and the information contained herein is provided on an
   "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING
   TASK FORCE DISCLAIMS 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.

Acknowledgement

   Funding for the RFC Editor function is currently provided by the
   Internet Society.