Internet Engineering Task Force
Internet-Draft

B. Halley
Nominum
E. Lewis
ARIN

August 10, 2003

Expires: February 10, 2004

Clarifying the Role of Wild Card Domains in the Domain Name System

<draft-ietf-dnsext-wcard-clarify-01.txt>

Status of this Memo

Abstract

The definition of wild cards is recast from the original in RFC 1034, in words that are more specific and in line with RFC 2119. This document is meant to supplement the definition in RFC 1034 and to alter neither the spirit nor intent of that definition.

1 Introduction

The first section of this document will give a crisp overview of what is begin defined, as well as the motivation rewording of an original document and making a change to bring the specification in line with implementations. Examples are included to help orient the reader.

Wild card domain names are defined in Section 4.3.3. of RFC 1034 as "instructions for synthesizing RRs." [RFC1034] The meaning of this is

that a specific, special domain name is used to construct responses in instances in which the query name is not otherwise represented in a zone.

A wild card domain name has a specific range of influence on query names (QNAMEs) within a given class, which is rooted at the domain name containing the wild card label, and is limited by explicit entries, zone cuts and empty non-terminal domains (see section 1.3 of this document).

Note that a wild card domain name has no special impact on the search for a query type (QTYPE). If a domain name is found that matches the QNAME (exact or a wild card) but the QTYPE is not found at that point, the proper response is that there is no data available. The search does not continue on to seek other wild cards that might match the QTYPE. To illustrate, a wild card owning an MX RR does not 'cover' other names in the zone that own an A RR. There are certain special case RR types that will be singled out for discussion, the SOA RR, NS RR, CNAME RR, and DNAME RR.

Why is this document needed? Empirical evidence suggests that the words in RFC 1034 are not clear enough. There exist a number of implementations that have strayed (each differently) from that definition. There also exists a misconception of operators that the wild card can be used to add a specific RR type to all names, such as the MX RR example cited above. This document is also needed as input to efforts to extend DNS, such as the DNS Security Extensions [RFC 2535]. Lack of a clear base specification has proven to result in extension documents that have unpredictable consequences. (This is true in general, not just for DNS.)

Another reason this clarification is needed is to answer questions regarding authenticated denial of existence, a service introduced in the DNS Security Extensions [RFC 2535]. Prior to the work leading up to this document, it had been feared that a large number of proof records (NXTs) might be needed in each reply because of the unknown number of potential wild card domains that were thought to be applicable. One outcome of this fear is a now discontinued document solving a problem that is now known not to exist. I.e., this clarification has the impact of defending against unwarranted protocol surgery. It is not "yet another" effort to just rewrite the early specifications for the sake of purity.

Although the effort to define the DNS Security Extensions has prompted this document, the clarifications herein relate to basic DNS only. No DNS Security Extensions considerations are mentioned in the document.

1.1 Document Limits

This document limits itself to reinforcing the concepts in RFC 1034. In the effort to do this, a few issues have been discussed that change parts of what is in RFC 1034. The discussions have been held within the DNS Extensions Working Group.

Briefly, the issues raised include:

- The lack of clarity in the definition of domain name existence
- Implications of a wild card domain name owning any of the following resource record sets: DNAME [RFC 2672], CNAME, NS, and SOA
- Whether RFC 1034 meant to allow special processing of CNAME RR's owned by wild card domain names

1.2 Existence

The notion that a domain name 'exists' will arise numerous times in this discussion. RFC 1034 raises the issue of existence in a number of places, usually in reference to non-existence and often in reference to processing involving wild card domain names. RFC 1034 contains algorithms that describe how domain names impact the preparation of an answer and does define wild cards as a means of synthesizing answers. Because of this a discussion on wild card domain names has to start with the issue of existence.

To help clarify the topic of wild cards, a positive definition of existence is needed. Complicating matters, though, is the realization that existence is relative. To an authoritative server, a domain name exists if the domain name plays a role following the algorithms of preparing a response. To a resolver, a domain name exists if there is any data available corresponding to the name. The difference between the two is the synthesis of records according to a wild card.

For the purposes of this document, the point of view of an authoritative server is adopted. A domain name is said to exist if it plays a role in the execution of the algorithms in RFC 1034.

1.3 An Example

For example, consider this wild card domain name: *.example. Any query name under example. is a candidate to be matched (answered) by this wild card, i.e., to have an response returned that is synthesized from the wild card's RR sets. Although any name is a candidate, not all queries will match.

To further illustrate this, consider this zone:

```
$ORIGIN example.
@       IN      SOA
                NS
                NS
*               TXT "this is a wild card"
                MX  10 mailhost.example.
host1           A   10.0.0.1
_ssh._tcp.host1 SRV
_ssh._tcp.host2 SRV
subdel          NS
```

The following queries would be synthesized from the wild card:

```
          QNAME=host3.example. QTYPE=MX, QCLASS=IN
                the answer will be a "host3.example. IN MX ..."
          QNAME=host3.example. QTYPE=A, QCLASS=IN
                the answer will reflect "no error, but no data"
                because there is no A RR set at '*'
```

The following queries would not be synthesized from the wild card:

```
          QNAME=host1.example., QTYPE=MX, QCLASS=IN
                because host1.example. exists
          QNAME=_telnet._tcp.host1.example., QTYPE=SRV, QCLASS=IN
                because _tcp.host1.example. exists (without data)
          QNAME=_telnet._tcp.host2.example., QTYPE=SRV, QCLASS=IN
                because host2.example. exists (without data)
          QNAME=host.subdel.example., QTYPE=A, QCLASS=IN
                because subdel.example. exists and is a zone cut
```

To the server, the following domains are considered to exist in the zone: *, host1, _tcp.host1, _ssh._tcp.host1, host2, _tcp.host2, _ssh._tcp.host2, and subdel. To a resolver, many more domains appear to exist via the synthesis of the wild card.

1.4 Empty Non-terminals

Empty non-terminals are domain names that own no data but have subdomains. This is defined in section 3.1 of RFC 1034:

```
#    The domain name space is a tree structure.  Each node and leaf on the
#    tree corresponds to a resource set (which may be empty).  The domain
#    system makes no distinctions between the uses of the interior nodes and
#    leaves, and this memo uses the term "node" to refer to both.
```

The parenthesized "which may be empty" specifies that empty non-terminals are explicitly recognized. According to the definition of existence in this document, empty non-terminals do exist at the server.

Carefully reading the above paragraph can lead to an interpretation that all possible domains exist - up to the suggested limit of 255 octets for a domain name [RFC 1035]. For example, www.example. may have an A RR, and as far as is practically concerned, is a leaf of the domain tree. But the definition can be taken to mean that sub.www.example. also exists, albeit with no data. By extension, all possible domains exist, from the root on down. As RFC 1034 also defines "an authoritative name error indicating that the name does not exist" in section 4.3.1, this is not the intent of the original document.

RFC1034's wording is to be clarified by adding the following paragraph:

> A node is considered to have an impact on the algorithms of 4.3.2 if it is a leaf node with any resource sets or an interior node, with or without a resource set, that has a subdomain that is a leaf

node with a resource set. A QNAME and QCLASS matching an existing node never results in a response return code of authoritative name error.

The terminology in the above paragraph is chosen to remain as close to that in the original document. The term "with" is a alternate form for "owning" in this case, hence "a leaf node owning resources sets, or an interior node, owning or not owning any resource set, that has a leaf node owning a resource set as a subdomain," is the proper interpretation of the middle sentence.

As an aside, an "authoritative name error" has been called NXDOMAIN in some RFCs, such as RFC 2136 [RFC 2136]. NXDOMAIN is the mnemonic assigned to such an error by at least one implementation of DNS. As this mnemonic is specific to implementations, it is avoided in the remainder of this document.

1.5 Terminology

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 the document entitled "Key words for use in RFCs to Indicate Requirement Levels." [RFC2119]

Requirements are denoted by paragraphs that begin with with the following convention: 'R'<sect>.<count>.

Quotations of RFC 1034 (as has already been done once above) are denoted by a '#' in the leftmost column.

2 Defining the Wild Card Domain Name

A wild card domain name is defined by having the initial label be:

0000 0001 0010 1010 (binary) = 0x01 0x2a (hexadecimal)

This defines domain names that may play a role in being a wild card, that is, being a source for synthesized answers. Domain names conforming to this definition that appear in queries and RDATA sections do not have any special role. These cases will be described in more detail in following sections.

R2.1 A domain name that is to be interpreted as a wild card MUST begin with a label of '0000 0001 0010 1010' in binary.

The first octet is the normal label type and length for a 1 octet long label, the second octet is the ASCII representation [RFC 20] for the '*' character. In RFC 1034, ASCII encoding is assumed to be the character encoding.

In the master file formats used in RFCs, a "*" is a legal representation

for the wild card label. Even if the "*" is escaped, it is still interpreted as the wild card when it is the only character in the label.

R2.2. A server MUST treat a wild card domain name as the basis of synthesized answers regardless of any "escape" sequences in the input format.

RFC 1034 and RFC 1035 ignore the case in which a domain name might be "the*.example.com." The interpretation is that this domain name in a zone would only match queries for "the*.example.com" and not have any other role.

Note: By virtue of this definition, a wild card domain name may have a subdomain. The subdomain (or sub-subdomain) itself may also be a wild card. E.g., *.*.example. is a wild card, so is *.sub.*.example. More discussion on this is given in Appendix A.

3 Defining Existence

As described in the Introduction, a precise definition of existence is needed.

R3.1 An authoritative server MUST treat a domain name as existing during the execution of the algorithms in RFC 1034 when the domain name conforms to the following definition. A domain name is defined to exist if the domain name owns data and/or has a subdomain that exists.

Note that at a zone boundary, the domain name owns data, including the NS RR set. At the delegating server, the NS RR set is not authoritative, but that is of no consequence here. The domain name owns data, therefore, it exists.

R3.2 An authoritative server MUST treat a domain name that has neither a resource record set nor an existing subdomain as non-existent when executing the algorithm in section 4.3.2. of RFC 1034.

A note on terminology. A domain transcends zones, i.e., all DNS data is in the root domain but segmented into zones of control. In this document, there are references to a "domain name" in the context of existing "in a zone." In this usage, a domain name is the root of a domain, not the entire domain. The domain's root point is said to "exist in a zone" if the zone is authoritative for the name. RR sets existing in a domain need not be owned by the domain's root domain name, but are owned by other domain names in the domain.

4 Impact of a Wild Card Domain In a Query Message or in an RDATA field

When a wild card domain name appears in a question, e.g., the query name is "*.example.", the response in no way differs from any other query. In other words, the wild card label in a QNAME has no special meaning,

and query processing will proceed using '*' as a literal query name.

R4.1 A wild card domain name acting as a QNAME MUST be treated as any other QNAME, there MUST be no special processing accorded it.

If a wild card domain name appears in the RDATA of a CNAME RR or any other RR that has a domain name in it, the same rule applies. In the instance of a CNAME RR, the wild card domain name is used in the same manner of as being the original QNAME. For other RR's, rules vary regarding what is done with the domain name(s) appearing in them, in no case does the wild card hold special meaning.

R4.2 A wild card domain name appearing in any RR's RDATA MUST be treated as any other domain name in that situation, there MUST be no special processing accorded it.

5 Impact of a Wild Card Domain On a Response

The description of how wild cards impact response generation is in RFC 1034, section 4.3.2. That passage contains the algorithm followed by a server in constructing a response. Within that algorithm, step 3, part 'c' defines the behavior of the wild card. The algorithm is directly quoted in lines that begin with a '#' sign. Commentary is interleaved.

There is a documentation issue deserving some explanation. The algorithm in RFC 1034, section 4.3.2. is not intended to be pseudo code, i.e., it's steps are not intended to be followed in strict order. The "algorithm" is a suggestion. As such, in step 3, parts a, b, and c, do not have to be implemented in that order.

Another issue needing explanation is that RFC 1034 is a full standard. There is another RFC, RFC 2672, which makes, or proposes an adjustment to RFC 1034's section 4.3.2 for the sake of the DNAME RR. RFC 2672 is a proposed standard. The dilemma in writing these clarifications is knowing which document is the one being clarified. Fortunately, the difference between RFC 1034 and RFC 2672 is not significant with respect to wild card synthesis, so this document will continue to state that it is clarifying RFC 1034. If RFC 2672 progresses along the standards track, it will need to refer to modifying RFC 1034's algorithm as amended here.

The context of part 'c' is that the search is progressing label by label through the QNAME. (Note that the data being searched is the authoritative data in the server, the cache is searched in step 4.) Step 3's part 'a' covers the case that the QNAME has been matched in full, regardless of the presence of a CNAME RR. Step 'b' covers crossing a cut point, resulting in a referral. All that is left is to look for the wild card.

Step 3 of the algorithm also assumes that the search is looking in the zone closest to the answer, i.e., in the same class as QCLASS and as close to the authority as possible on this server. If the zone is not

the authority, then a referral is given, possibly one indicating lameness.

```
#         c. If at some label, a match is impossible (i.e., the
#            corresponding label does not exist), look to see if a
#            the "*" label exists.
```

The above paragraph refers to finding the domain name that exists in the zone and that most encloses the QNAME. Such a domain name will mark the boundary of candidate wild card domain names that might be used to synthesize an answer. (Remember that at this point, if the most enclosing name is the same as the QNAME, part 'a' would have recorded an exact match.) The existence of the enclosing name means that no wild card name higher in the tree is a candidate to answer the query.

Once the closest enclosing node is identified, there's the matter of what exists below it. It may have subdomains, but none will be closer to the QNAME. One of the subdomains just might be a wild card. If it exists, this is the only wild card eligible to be used to synthesize an answer for the query. Even if the closest enclosing node conforms to the syntax rule in section 2 for being a wild card domain name, the closest enclosing node is not eligible to be a source of a synthesized answer.

The only wild card domain name that is a candidate to synthesize an answer will be the "*" subdomain of the closest enclosing domain name. Three possibilities can happen. The "*" subdomain does not exist, the "*" subdomain does but does not have an RR set of the same type as the QTYPE, or it exists and has the desired RR set.

For the sake of brevity, the closest enclosing node can be referred to as the "closest encloser." The closest encloser is the most important concept in this clarification. Describing the closest encloser is a bit tricky, but it is an easy concept.

To find the closest encloser, you have to first locate the zone that is the authority for the query name. This eliminates the need to be concerned that the closest encloser is a cut point. In addition, we can assume too that the query name does not exist, hence the closest encloser is not equal to the query name. We can assume away these two cases because they are handled in steps 2, 3a and 3b of section 4.3.2.'s algorithm.

What is left is to identify the existing domain name that would have been up the tree (closer to the root) from the query name. Knowing that an exact match is impossible, if there is a "*" label descending from the unique closest encloser, this is the one and only wild card from which an answer can be synthesized for the query.

To illustrate, using the example in section 1.2 of this document, the following chart shows QNAMEs and the closest enclosers. In Appendix A there is another chart showing unusual cases.

QNAME	Closest Encloser	Wild Card Source

```
     host3.example.                 example.              *.example.
     _telnet._tcp.host1.example.    _tcp.host1.example.   no wild card
     _telnet._tcp.host2.example.    host2.example.        no wild card
     _telnet._tcp.host3.example.    example.              *.example.
     _chat._udp.host3.example.      example.              *.example.
```

Note that host1.subdel.example. is in a subzone, so the search for it ends in a referral in part 'b', thus does not enter into finding a closest encloser.

The fact that a closest encloser will be the only superdomain that can have a candidate wild card will have an impact when it comes to designing authenticated denial of existence proofs.

```
#            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.
```

The above passage says that if there is not even a wild card domain name to match at this point (failing to find an explicit answer elsewhere), we are to return an authoritative name error at this point. If we were following a CNAME, the specification is unclear, but seems to imply that a no error return code is appropriate, with just the CNAME RR (or sequence of CNAME RRs) in the answer section.

```
#            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.
```

This final paragraph covers the role of the QTYPE in the process. Note that if no resource record set matches the QTYPE the result is that no data is copied, but the search still ceases ("Go to step 6."). In the following section, a suggested change is made to this, under the heading "CNAME RRs at a Wild Card Domain Name."

6 Considerations with Special Types

For the purposes of this section, "special" means that a record induces processing at the server beyond simple lookup. The special types in this section are SOA, NS, CNAME, and DNAME. SOA is special because it is used as a zone marker and has an impact on step 2 of the algorithm in 4.3.2. NS denotes a cut point and has an impact on step 3b. CNAME redirects the query and is mentioned in steps 3a and 3b. DNAME is a "CNAME generator."

6.1 SOA RR's at a Wild Card Domain Name

If the owner of an SOA record conforms to the basic rules of owning an SOA RR (meaning it is the apex of a zone) the impact on the search algorithm

is not in section 3c (where records are synthesized) as would be expected. The impact is really in step 2 of the algorithm, the choice of zone.

We are no longer talking about whether or not an SOA RR can be synthesized in a response because we are shifting attention to step 2. We are now talking about what it means for a name server to synthesize a zone for a response. To date, no implementation has done this. Thinking ahead though, anyone choosing to pursue this would have to be aware that a server would have to be able to distinguish between queries for data it will have to synthesize and queries that ought to be treated as if they were prompted by a lame delegation.

It is not a protocol error to have an SOA RR owned by a wild card domain name, just as it is not an error to have zone name be syntactically equivalent to a domain name. However, this situation requires careful consideration of how a server chooses the appropriate zone for an answer. And an SOA RR is not able to be synthesized as in step 3c.

6.2 NS RR's at a Wild Card Domain Name

Complimentary to the issue of an SOA RR owned by a wild card domain name is the issue of NS RR's owned by a wild card domain name. In this instance, each machine being referred to in the RDATA of the NS RR has to be able to understand the impact of this on step 2, the choosing of the authoritative zone.

Referring to the same machine in such a NS RR will probably not work well. This is because the server may become confused as to whether the query name ought to be answered by the zone owning the NS RR in question or a synthesized zone. (It isn't known in advance that the query name will invoke the wild card synthesis.)

The status of other RR's owned by a wild card domain name is the same as if the owner name was not a wild card domain name. I.e., when there is a NS RR at a wild card domain name, other records are treated as being below the zone cut.

Is it not a protocol error to have a NS RR owned by a wild card domian name, complimentary to the case of a SOA RR. However, for this to work, an implementation has to know how to synthesize a zone.

6.3 CNAME RR's at a Wild Card Domain Name

The issue of CNAME RR's owned by wild card domain names has prompted a suggested change to the last paragraph of step 3c of the algorithm in 4.3.2. The changed text is this:

> If the "*" label does exist and 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, set the owner of the CNAME RR to be QNAME, and then change QNAME to the canonical name in the CNAME RR, and go back to step 1.

> If the "*" label does exist and either QTYPE is CNAME or the data at the node is not a CNAME, then 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.

Apologies if the above isn't clear, but an attempt was made to stitch together the passage using just the phrases in section 3a and 3c of the algorithm so as to preserve the original flavor.

In case the passage as suggested isn't clear enough, the intent is to make "landing" at a wild card name and finding a CNAME the same as if this happened as a result of a direct match. I.e., Finding a CNAME at the name matched in step 3c is supposed to have the same impact as finding the CNAME in step 3a.

6.4 DNAME RR's at a Wild Card Domain Name

The specification of the DNAME RR, which is at the proposed level of standardization, is not as mature as the full standard in RFC 1034. Because of this, or the reason for this is, there appears to be a host of issues with that definition and it's rewrite of the algorithm in 4.3.2. For the time being, when it comes to wild card processing issues, a DNAME can be considered to be a CNAME synthesizer. A DNAME at a wild card domain name is effectively the same as a CNAME at a wild card domain name.

7 Security Considerations

This document is refining the specifications to make it more likely that security can be added to DNS. No functional additions are being made, just refining what is considered proper to allow the DNS, security of the DNS, and extending the DNS to be more predictable.

8 References

Normative References

[RFC 20] ASCII Format for Network Interchange, V.G. Cerf, Oct-16-1969
[RFC 1034] Domain Names - Concepts and Facilities, P.V. Mockapetris, Nov-01-1987
[RFC 1035] Domain Names - Implementation and Specification, P.V Mockapetris, Nov-01-1987
[RFC 2119] Key Words for Use in RFCs to Indicate Requirement Levels, S Bradner, March 1997

Informative References

[RFC 2136] Dynamic Updates in the Domain Name System (DNS UPDATE), P. Vixie, Ed., S. Thomson, Y. Rekhter, J. Bound, April 1997
[RFC 2535] Domain Name System Security Extensions, D. Eastlake, March 1999

[RFC 2672] Non-Terminal DNS Name Redirection, M. Crawford, August 1999

9 Others Contributing to This Document

Others who have directly caused text to appear in the document: Paul Vixie and Olaf Kolkman. Many others have indirect influences on the content.

10 Editors

Name: Bob Halley
Affiliation: Nominum, Inc.
Address: 2385 Bay Road, Redwood City, CA 94063 USA
Phone: +1-650-381-6016
EMail: Bob.Halley@nominum.com

Name: Edward Lewis
Affiliation: ARIN
Address: 3635 Concorde Pkwy, Suite 200, Chantilly, VA 20151 USA
Phone: +1-703-227-9854
Email: edlewis@arin.net

Appendix A: Subdomains of Wild Card Domain Names

In reading the definition of section 2 carefully, it is possible to rationalize unusual names as legal. In the example given, *.example. could have subdomains of *.sub.*.example. and even the more direct *.*.example. (The implication here is that these domain names own explicit resource records sets.) Although defining these names is not easy to justify, it is important that implementions account for the possibility. This section will give some further guidence on handling these names.

The first thing to realize is that by all definitions, subdomains of wild card domain names are legal. In analyzing them, one realizes that they cause no harm by their existence. Because of this, they are allowed to exist, i.e., there are no special case rules made to disallow them. The reason for not preventing these names is that the prevention would just introduce more code paths to put into implementations.

The concept of "closest enclosing" existing names is important to keep in mind. It is also important to realize that a wild card domain name can be a closest encloser of a query name. For example, if *.*.example. is defined in a zone, and the query name is a.*.example., then the closest enclosing domain name is *.example. Keep in mind that the closest encloser is not eligible to be a source of synthesized answers, just the subdomain of it that has the first label "*".

To illustrate this, the following chart shows some matches. Assume that the names *.example., *.*.example., and *.sub.*.example. are defined in the zone.

QNAME	Closest Encloser	Wild Card Source
a.example.	example.	*.example.
b.a.example.	example.	*.example.
a.*.example.	*.example.	*.*.example.
b.a.*.example.	*.example.	*.*.example.
b.a.*.*.example.	*.*.example.	no wild card
a.sub.*.example.	sub.*.example.	*.sub.*.example.
b.a.sub.*.example.	sub.*.example.	*.sub.*.example.
a.*.sub.*.example.	*.sub.*.example.	no wild card
*.a.example.	example.	*.example.
a.sub.b.example.	example.	*.example.

Recall that the closest encloser itself cannot be the wild card. Therefore the match for b.a.*.*.example. has no applicable wild card.

Finally, if a query name is sub.*.example., any answer available will come from an exact name match for sub.*.example. No wild card synthesis is performed in this case.

Full Copyright Statement

Acknowledgement

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

_ _