DNSEXT Working Group
INTERNET DRAFT
Expiration Date: April 2005

E. Lewis
NeuStar

October 2004

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

draft-ietf-dnsext-wcard-clarify-03.txt

Status of this Memo

Copyright Notice

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 not to significantly alter the spirit or intent of that definition.

1 Introduction

In RFC 1034 [RFC1034], sections 4.3.2 and 4.3.3 describe the synthesis of answers from special records called wildcards. The original definitions are incomplete. This document clarifies and describes

the wildcard synthesis by adding to the discussion and making limited modifications. Modifications are made only where necessary to close inconsistencies that have led to interoperability issues.

1.1 Motivation

Over time many implementations have diverged in different ways from the original definition, or at least what had been intended. Although there is clearly a need to clarify the original documents in light of this, the impetus for this document lay in the engineering of the DNS security extensions [RFC TBD]. With an unclear definition of wildcards the design of authenticated denial became entangled.

Although this document is motivated by DNSSEC and the need to have a separate document passed for the sake of DNSSEC, other motivations have risen. The renewed understanding of wildcards gained is worthy of being documented.

1.2 The Original Definition

This document is intended to not make changes. To reinforce this, sections of RFC 1034 are repeated verbatim for convenience of the reader, to help in comparison of old and new text.

There are a few passages which are changed. This may seem to contradict the goal of not changing the original specification, but the changes herein are required because of inconsistencies with the wording in RFC 1034.

The beginning of the discussion ought to start with the definition of the term "wildcard" as it appears in RFC 1034, section 4.3.3.

```
# In the previous algorithm, special treatment was given to RRs with owner
# names starting with the label "*".  Such RRs are called wildcards.
# Wildcard RRs can be thought of as instructions for synthesizing RRs.
# When the appropriate conditions are met, the name server creates RRs
# with an owner name equal to the query name and contents taken from the
# wildcard RRs.
```

This passage appears after the algorithm in which they are used is presented. The terminology is not consistent, the word "wildcard" is clearly defined to be a resource record. In the next sentence the term is shifted to be an adjective, the first step on the path to overloading the term. Wildcard has also been used to refer to domain names that begin with a "*".

1.3 The Clarification

The clarification effort can be divided into three sections. One is the use of new terminology to better describe wildcards. Changes to words in RFC 1034 that have resulted by discovering conflicting

concepts are presented. Descriptions of special type records in the context of being wildcards is discussed.

1.3.1 New Terms

The term "wildcard" has become so overloaded it is virtually useless as a description. A few new terms will be introduced to be more descriptive. The new terms that will be introduced are:

Asterisk Label - a label consisting of an asterisk ("*") and no other characters.

Wild Card Domain Name - a domain name whose least significant label (first when reading left to right) is an asterisk label. Other labels might also be asterisk labels.

Source of Synthesis - a Wild Card Domain Name when it is consulted in the final paragraph of step 3, part c of RFC 1034's 4.3.2 algorithm.

Closest Encloser - in RFC 1034's 4.3.2 algorithm, the name at which the last match was possible in step 3, part c. This is the longest sequence of exactly matching labels from the root downward in both the sought name (QNAME) and in the zone being examined.

Label Match - two labels are equivalent if the label type and label length are the same bit sequence and if the name is the label is equivalent bit wise after down casing all of the ASCII characters. [Ed note: do we still call them ASCII?]

These terms will be more fully described as needed later. These terms will be used to describe a few changes to the words in RFC 1034. A summary of the changes appear next and will be fully covered in later sections.

1.3.2 Changed Text

The definition of "existence" is changed, superficially, to exclude empty domains that have no subdomains with resource records. This change will not be apparent to implementations, it is needed to make descriptions more concise.

In RFC 1034, there is text that seems to bar having two Asterisk Labels in a Wild Card Domain Name. There is no further discussion, no prescribed error handling, nor enforcement described. In this document, the use of such names will be discouraged, but implementations will have to account for the possibility of such a name's use.

The actions when a Source of Synthesis owns a CNAME RR are changed to mirror the actions if an exact match name owns a CNAME RR. This is an addition to the words in RFC 1034, section 4.3.2, step 3, part c.

1.3.3 Considerations with Special Types

This clarification will describe in some detail the semantics of wildcard CNAME RRs, wildcard NS RRs, wildcard SOA RR's, wildcard DNAME RRs [RFC wxyz], and empty, non-terminal wildcards. Understanding these types in the context of wildcards has been clouded because these types incur special processing if they are the result of an exact match.

By the definition in RFC 1034, there can be no empty, non-terminal "wildcards", but in the algorithm, it is possible that an empty non-terminal is sought as the potential owner of a "wildcard." This is one example of why the ordering of the discussion in RFC 1034 is confusing.

1.4 Standards 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]

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

2 "Wildcard"

The context of the wildcard concept involves the algorithm by which a name server prepares a response (in RFC 1034's section 4.3.2) and the way in which a resource record (set) is identified as being a source of synthetic data (section 4.3.3).

Tackling the latter first, there are two objectives in defining a means to identify a resource record set as a source of synthesis. First is the desire to maintain all DNS data in a consistent manner. Avoiding the need for implementations to have many internal data structures is a good thing. Not that this means limiting quantity, but rather types of data. The second objective impacts interoperability, that is a master server of one implementation has to be able to send the synthesis instructions to the slaves. Although there are alternatives to the use of zone transfers via port 53, a truly interoperable record synthesis approach has to be able to insert the synthesis instructions into a zone transfer.

The objectives in describing the synthesis of records in the context of the name server algorithm include knowing when to employ the process of synthesis and how the synthesis is carried out.

2.1 Identifying a wildcard

To provide a more accurate description of "wildcards", the definition has to start with a discussion of the domain names that appear as owners.

2.1.1 Wild Card Domain Name and Asterisk Label

A "Wild Card Domain Name" is defined by having its initial label be:

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

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.

A descriptive name of a label equaling that value is an "Asterisk Label."

RFC 1034's definition of wildcard would be "a resource record owned by a Wild Card Domain Name." This is mentioned to help maintain some orientation between this clarification and RFC 1034. Keep in mind, that in "Clarifications to the DNS Specification" [RFC 2181] the name of the basic unit of DNS data became the resource record set (RRSet) and not the resource record.

2.1.2 Variations on Wild Card Domain Names

RFC 1034 and RFC 1035 do not explicitly mention the case in which a domain name might be something like "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. An asterisk ('*') occurring other than as the sole character in a label is simply a character forming part of the label and has no special meaning. This is not an Asterisk Label, simply a label an asterisk in it. The same is true for "**.example.com." and "*the.example.com."

[Ed note: the above paragraph reads too strong. The intent ought to be that such names do not fall under the rules of wildcards. The intent is not to bar any future attempts to define other forms of synthesis - nor is the intent to encourage them.]

The interpretation of a wild card domain specification which is not a leaf domain is not clearly defined in RFC 1034. E.g., sub.*.example., is not discussed, not barred. In wanting to minimize changes from the original specification, such names are permitted. Although "sub.*.example." is not a Wild Card Domain Name, "*.example." is.

RRSets used to synthesize records can be owned by a Wild Card Domain Name that has subdomains.

2.1.3 Non-terminal Wild Card Domain Names

In section 4.3.3, the following is stated:

```
#   ........................  The owner name of the wildcard RRs is of
#   the form "*.<anydomain>", where <anydomain> is any domain name.
#   <anydomain> should not contain other * labels.....................
```

This covers names like "*.foo.*.example." The pre-RFC2119 wording uses "should not" which has an ambiguous meaning. The specification does not proscribe actions upon seeing such a name, such as whether or not a zone containing the name should fail to be served. What if a dynamic update (RFC2136) requested to add the name to the zone? The failure semantics are not defined.

The recommendation is that implementations ought to anticipate the appearance of such names but generally discourage their use in operations. No standards statement, such as "MAY NOT" or "SHOULD NOT" is made here.

The interpretation of this is, when seeking a Wild Card Domain Name for the purposes of record synthesis, an implementation ought not to check the domain name for subdomains.

It is possible that a Wild Card Domain Name is an empty non-terminal. (See the upcoming sections on empty non-terminals.) In this case, the lookup will terminate as would any empty non-terminal match.

2.2 Existence Rules

The notion that a domain name 'exists' arises numerous times in discussions about the wildcard concept. RFC 1034 raises the issue of existence in a number of places, usually in reference to non-existence and in reference to processing involving wildcards. RFC 1034 contains algorithms that describe how domain names impact the preparation of an answer and does define wildcards as a means of synthesizing answers. Because of this a discussion on wildcards needs to cover a definition 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 wildcard.

For the purposes of this document, the point of view of an authoritative server is more interesting. A domain name is said to

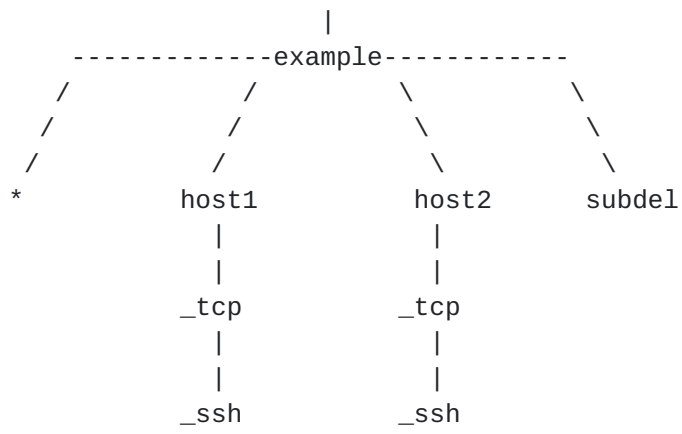
exist if it plays a role in the execution of the algorithms in RFC 1034.

2.2.1. An Example

To illustrate what is meant by existence consider this complete zone:

```
$ORIGIN example.
example.                  3600 IN  SOA   <SOA RDATA>
example.                  3600     NS    ns.example.com.
example.                  3600     NS    ns.example.net.
*.example.                3600     TXT   "this is a wild card"
*.example.                3600     MX    10 host1.example.
host1.example.            3600     A     192.0.4.1
_ssh._tcp.host1.example.  3600     SRV  <SRV RDATA>
_ssh._tcp.host2.example.  3600     SRV  <SRV RDATA>
subdel.example.           3600     NS   ns.example.com.
subdel.example.           3600     NS   ns.example.net.
```

A look at the domain names in a tree structure is helpful:



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 '*.example.'

QNAME=foo.bar.example. QTYPE=TXT, QCLASS=IN
the answer will be "foo.bar.example. IN TXT ..."
because bar.example. does not exist, but the wildcard does.

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

```
QNAME=host1.example., QTYPE=MX, QCLASS=IN
     because host1.example. exists

QNAME=ghost.*.example., QTYPE=MX, QCLASS=IN
     because *.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, all of the domains in the tree exist. The resolver will get answers to some names off the tree, thanks to synthesis.

2.2.2 Empty Non-terminals

Empty non-terminals are domain names that own no resource records but have subdomains which do. 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.

Pedantically 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.

2.2.3 Yet Another Definition of Existence

RFC1034's wording is clarified by 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 code of authoritative name error (RCODE==3).

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", response code (RCODE) 3, 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.

Summarizing the discussion on existence in non-RFC1034 words:

> An authoritative server is to 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 or has a subdomain that exists, or both.

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.

2.3 When does a Wild Card Domain Name not own a wildcard (record)

When a Wild Card Domain Name appears in a message's query section, no special processing occurs. Asterisk Labels in such a context only Label Matches other Asterisk Labels in the existing zone tree when the 4.3.2 algorithm is being followed.

When a Wild Card Domain Name appears in the resource data of a record, no special processing occurs. An Asterisk Label in that context literally means just an asterisk.

3. 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.

3.1 Step 2

Step 2 of the RFC 1034's section 4.3.2 reads:

```
#    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.
```

In this step, the most appropriate zone for the response is chosen. There are two reasons to repeat this. One is that this means all of step 3 is done within the context of a zone, which will constrain the discussion. The other is the though behind synthesizing entire zones and the use of Wild Card Domain Names to do so.

3.2 Step 3

Step 3 is dominated by three parts, labelled a, b, and c. But the beginning of the Step is important and needs explanation.

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

The word matching in this care refers to Label Matching. The concept is based in the view of the zone as the tree of existing names. The Query Name is considered to be an ordered sequence of labels - as if the name were a path from the root to the owner of the desired data.

The process of Label Matching ends in one of three choices, the parts a, b, and c. Once one of the parts is chosen, the other parts are not considered. (E.g., do not execute part c and then change the execution path to finish in part b.) The process of Label Matching is also done independent of the Query Type.

Parts a and b are not an issue for this clarification as they do not relate to record synthesis. Part a generally covers a situation in which all of the labels in the search (query) name have been matched down the tree, e.g., the sought name exists as an exact Label Match. Part b generally covers a situation in which any label in the sought name Label Matches a tree label and the tree label has a NS RRSet.

3.3 Part 'c'

The context of part 'c' is that the process of Label Matching the labels in the sought name has resulted in a situation in which there is nothing corresponding in the tree. It is as if the lookup has "fallen off the tree."

```
#         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.
```

To help describe the process of looking "to see is a the [sic] label exists" a term has been coined to describe the last label matched. The term is "Closest Encloser."

3.3.1 Closest Encloser and the Source of Synthesis

The "Closest Encloser" is the node in the zone's tree of existing domain names that is has the most matching labels with the sought name. Each match is a "Label Match" and the order of the labels is also the same. The Closest Encloser is an existing name in the zone, it may be an empty non-terminal, it may even be a Wild Card Domain Name itself. In no circumstances is the Closest Encloser the used to synthesize records though.

A "Source of Synthesis" is defined in the context of a lookup process as the Wild Card Domain Name immediately descending from the Closest Encloser provided the Wild Card Domain Name exists. A Source of Synthesis does not guarantee having a RRSet to use for synthesis, a Source of Synthesis may even be an empty non-terminal.

If a Source of Synthesis exists, it will be the Wild Card Domain Name that is identified by an Asterisk Label below the Closest Encloser. E.g., "<Asterisk Label>.<Closest Encloser> or "*.<Closest Encloser>." If the Source of Synthesis does not exist (not on the domain tree), there will be no wildcard synthesis

The important concept is that for any given lookup process, there is at most one place at which wildcard synthetic records can be obtained. If the Source of Synthesis does not exist, the lookup terminates, the lookup does not look for other wildcard records.

Other terms have been coined on the mailing list in the past. E.g.,

it has been said that existing names block the application of wildcard records. This is still an appropriate viewpoint, but replacing this notion with the Closest Encloser and Source of Synthesis the depiction of the wildcard process is clearer.

3.3.2 Closest Encloser and Source of Synthesis Examples

To illustrate, using the example zone in section 2.2.1 of this document, the following chart shows QNAMEs and the closest enclosers.

QNAME	Closest Encloser	Source of Synthesis
host3.example.	example.	*.example.
_telnet._tcp.host1.example.	_tcp.host1.example.	no source
_telnet._tcp.host2.example.	host2.example.	no source
_telnet._tcp.host3.example.	example.	*.example.
_chat._udp.host3.example.	example.	*.example.

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 pre-calculated authenticated denial of existence proofs.

3.3.3 Non-existent Source of Synthesis

In RFC 1034:

```
#           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 is clear, evidenced by the lack of discussion and mis-implementation of it over the years. It is included for completeness only. (No attempt is made to re-interpret it lest a mistake in editing leads to confusion.)

3.3.4 Type Matching

RFC 1034 concludes part c with this:

```
#           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 lookup process.

Based on implementation feedback and similarities between step a and step c a change to this passage a change has been made.

The change is to add the following text:

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

This is essentially the same text in step a covering the processing of CNAME RRSets.

4. Considerations with Special Types

Five types of RRSets owned by a Wild Card Domain Name have caused confusion. Four explicit types causing confusion are SOA, NS, CNAME, DNAME, and the fifth type - "none."

4.1. SOA RR's at a Wild Card Domain Name

A Wild Card Domain Name owning an SOA RRSet means that the domain is at the root of the zone (apex). The domain can not be a Source of Synthesis because that is, but definition, a descendent node (of the Closest Encloser) and a zone apex is at the top of the zone.

Although a Wild Card Domain Name can not be a Source of Synthesis, there is no reason to forbid the ownership of an SOA RRSet. This means that zones with names like "*.<Parent Zone>.", and even "*.<Parent Sublabels>.<Parent Zone>."

Step 2 (section 3.1) does not provide a means to specify a means to synthesize a zone. Therefore, according to the rules there, the only way in which a zone that has a name which is a Wild Card Domain Name is if the QNAME is in a domain below the zone's name.

E.g., if *.example. has an SOA record, then only a query like QNAME=*.example., QTYPE=A, QCLASS=IN would see it. As another example, a QNAME of www.*.example. would also result in passing through the zone.

4.2. NS RR's at a Wild Card Domain Name

The semantics of a Wild Card Domain Name ownership of a NS RRSet has been unclear. Looking through RFC 1034, the recommendation is to have the NS RRSet act the same a any non-special type, e.g., like an A RR.

If the NS RRSet in question is at the top of the zone, i.e., the name also owns an SOA RRSet, the QNAME equals the zone name. This would trigger part 'a' of Step 3.

In any other case, the Wild Card Domain Name owned NS RRSet would be the only RRSet (prior to changes instituted by DNSSEC) at the node by DNS rules. If the QNAME equals the Wild Card Domain Name or is a subdomain of it, then the node would be considered in part 'b' of Step 3.

Note that there is no synthesis of records in the authority section because part 'b' does not account for synthesis. The referral returned would have the Wild Card Domain Name in the authority section, unchanged.

If the QNAME is not the same as the Wild Card Domain Name nor a subdomain of it, then part 'c' of Step 3 has been triggered. Once part 'c' is entered, there is no reverting to part 'b' - i.e., once an NS RRSet is synthesized it does not mean that the server has to consider the name delegated away. I.e., the server is not synthesizing a record (the NS RRSet) that means the server does not have the right to synthesize.

4.3. 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 appears in section 3.3.4 of this document.

4.4. 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 a number 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.

4.5 Empty Non-terminal Wild Card Domain Name

If a Source of Synthesis is an empty non-terminal, then the response will be one of no error in the return code and no RRSet in the answer section.

5. 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.

6. 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

7. Others Contributing to This Document

Others who have been editors of this document: Bob Halley and Robert Elz. Others who have directly caused text to appear in the document: Paul Vixie and Olaf Kolkman. Many others have indirect influences on the content.

8. Editor

Name:	Edward Lewis
Affiliation:	NeuStar
Address:	tbd
Phone:	tbd
Email:	tbd (please send comments to namedroppers)

Comments on this document can be sent to the editors or the mailing list for the DNSEXT WG, namedroppers@ops.ietf.org.

9. Trailing Boilerplate

"AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr. The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

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

Expiration

This document expires on or about 11 April 2005.