DNS Extensions Working Group
Internet-Draft
Expires: June 1, 2005

G. Sisson
B. Laurie
Nominet
December 1, 2004

Derivation of DNS Name Predecessor and Successor
draft-sisson-dnsext-dns-name-p-s-01

Status of this Memo

Copyright Notice

Abstract

This document describes a method for deriving the canonically-ordered predecessor and successor of a DNS name. This is expected to be useful for real-time NSEC resource record synthesis, which may be used in alterative implementations of DNSSEC-enabled DNS servers.

Table of Contents

1. Introduction

One of the proposals for avoiding the exposure of zone information while deploying DNSSEC is dynamic NSEC synthesis. This technique is described in [I-D.ietf-dnsext-dnssec-trans] and [I-D.weiler-dnsext-dnssec-online-signing], and involves the generation of NSEC RRs which just span the query name for non-existent owner names. In order to do this, the DNS names which would occur just prior to and just following a given query name must be calculated in real time, as maintaining a list of all possible owner names that might occur in a zone would normally be prohibitive.

Section 6.1 of [I-D.ietf-dnsext-dnssec-records] defines canonical DNS name order. This document does not amend or modify this definition. However, the derivation of immediate predecessor and successor, while trivial, is non-obvious. Accordingly, the method is described here as an aid to implementors and a reference to other interested parties.

2. Derivation of DNS Name Predecessor

This derivation assumes that all upper-case US-ASCII letters in the DNS name have already been replaced by their corresponding lower-case equivalents.

To derive the immediate predecessor of a DNS name:

1. If the DNS name is the same as the owner name of the apex, prepend the DNS name repeatedly with labels of the maximum length possible consisting of octets of the maximum sort value (e.g. 0xff) until the DNS name is the maximum length possible; otherwise continue to the next step.

2. If the least significant (left-most) label consists of a single octet of the minimum sort value (e.g. 0x00), remove that label;

otherwise continue to the next step.

3. If the least significant (right-most) octet in the least significant (left-most) label is the minimum sort value, remove that octet and continue with step 5.

4. Decrement the value of the least significant (right-most) octet, skipping any values which correspond to upper-case US-ASCII letters, and then append the label with as many octets as possible of the maximum sort value. Continue to the next step.

5. Prepend the DNS name repeatedly with labels of as long a length as possible consisting of octets of the maximum sort value until

the DNS name is the maximum length possible.

3. Derivation of DNS Name Successor

This derivation assumes that all upper-case US-ASCII letters in the DNS name have already been replaced by their corresponding lower-case equivalents.

To derive the immediate successor of a DNS name:

1. If the DNS name is two or more octets shorter than the maximum DNS name length, prepend the DNS name with a label containing a single octet of the minimum sort value (e.g. 0x00); otherwise continue to the next step.

2. If the DNS name is one or more octets shorter than the maximum DNS name length and the least significant (left-most) label is one or more octets shorter than the maximum label length, append an octet of the minimum sort value to the least significant label; otherwise continue to the next step.

3. Increment the value of the least significant (right-most) octet in the least significant (left-most) label that is less than the maximum sort value (e.g. 0xff), skipping any values which correspond to upper-case US-ASCII letters, and then remove any octets to the right of that one. If all octets in the label are the maximum sort value, then continue to the next step.

4. Remove the least significant (left-most) label. If the DNS name is now the same as the owner name of the apex, do nothing. (This will occur only if the DNS name was the maximum possible in canonical DNS name order, and thus has wrapped to the apex.) Otherwise repeat starting at Step 2.

4. Notes

4.1 Case Considerations

Section 3.5 of [RFC1034] specifies that "while upper and lower case letters are allowed in [DNS] names, no significance is attached to the case". Additionally, Section 6.1 of [I-D.ietf-dnsext-dnssec-records] states that when determining canonical DNS name order, "upper case US-ASCII letters are treated as if they were lower case US-ASCII letters". Consequently, values corresponding to US-ASCII upper-case letters must be skipped when decrementing and incrementing octets in the derivations described in

Section 2 and Section 3.

The following pseudo-code is illustrative:

Decrementing the value of an octet:

```
if (octet == '[')       // '[' is just after upper-case 'Z'
        octet = '@';    // '@' is just prior to upper-case 'A'
else
        octet--;
```

Incrementing the value of an octet:

```
if (octet == '@')       // '@' is just prior to upper-case 'A'
        octet = '[';    // '[' is just after upper-case 'Z'
else
        octet++;
```

4.2 Choice of Range

[RFC2181] makes the clarification that "any binary string whatever can be used as the label of any resource record". Consequently the minimum sort value may be set as 0x00 and the maximum sort value as 0xff, and the range of possible values will be any DNS name which contains octets of any value other than those corresponding to upper-case US-ASCII letters.

However, if all owner names in a zone are in the letter-digit-hyphen, or LDH, format specified in [RFC1034], it may be desirable to restrict the range of possible values to DNS names containing only LDH values. This has the effect of:

1. making the output of tools such as `dig' and `nslookup' less potentially confusing;

2. minimising the impact that NSEC RRs containing DNS names with non-LDH values (or non-printable values) might have on faulty DNS resolver implementations; and

3. preventing the possibility of results which are wild card DNS names (see Section 4.3).

This may be accomplished by using a minimum sort value of 0x1f (US-ASCII character `-') and a maximum sort value of 0x7a (US-ASCII character lower-case `z'), and then skipping non-LDH, non-lower-case values when incrementing or decrementing octets.

4.3 Wild Card Considerations

Neither derivation avoids the possibility that the result may be a DNS name containing a wild card label, i.e. a label containing a single octet with the value 0x2a (US-ASCII character `*'). With additional tests, wild card DNS names may be explicitly avoided; alternatively, if the range of octet values can be restricted to those corresponding to letter-digit-hyphen, or LDH, characters (see Section 4.2), such DNS names will not occur.

Note that it is improbable that a result which is a wild card DNS name will occur unintentionally; even if one does occur either as the owner name of, or in the RDATA of an NSEC RR, it is treated as a literal DNS name with no special meaning.

4.4 Potential Optimisations

4.4.1 Omission of Step

When the derivation of immediate predecessor is used only for the synthesis of NSEC RRs, step 1 of the derivation may be omitted as the existence of the owner name of the apex should never need to be denied. This eliminates one condition that would otherwise always be tested during the derivation of the immediate predecessor.

4.4.2 Restriction of Effective Maximum DNS Name Length

[RFC1034] specifies that "the total number of octets that represent a [DNS] name (i.e., the sum of all label octets and label lengths) is limited to 255", including the null (zero-length) label which represents the root. For the purpose of deriving the immediate predecessor and successor during NSEC RR synthesis, the maximum DNS name length may be effectively restricted to the length of the longest DNS name in the zone. This will minimise the size of responses containing synthesised NSEC RRs.

Note that this optimisation will have the effect of revealing information about the longest name in the zone. Moreover, when the contents of the zone changes, e.g. during dynamic updates and zone transfers, care must be taken to ensure that the effective maximum DNS name length agrees with the new contents.

A modified version of this optimisation will realise most of its benefit while mitigating these exposures: if the length of unqulified owner names of empty non-terminals in a zone is restricted to 64 octets in wire format, then the effective maximum DNS name length may be restricted to 64 + the length of the owner name of the apex. This will prevent the discovery of the longest single label in the zone,

which is of more concern to most zone operators who are concerned about owner name elaboration.

5. Examples

In the following examples:

the owner name of the apex is "example.com.";

the range of octet values is 0x00 - 0xff excluding values corresponding to upper-case US-ASCII letters; and

non-printable octet values are expressed as three-digit decimal numbers preceded by a backslash (as specified in Section 5.1 of [RFC1035]).

5.1 Examples of Immediate Predecessors

Example of typical case:

```
x  = foo.example.com.

x' = \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255.\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255.\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255255.fon\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255255.example.com.
```

or, in alternate notation:

```
     \255{49}.\255{63}.\255{63}.fon\255{60}.example.com.
```

where {n} represents the number of repetitions of an octet.

Example where least significant (left-most) label of DNS name consists of a single octet of the minimum sort value:

```
x  = \000.foo.example.com.

x' = foo.example.com.
```

Example where least significant (right-most) octet of least significant (left-most) label has the minimum sort value:

```
x  = foo\000.example.com.

x' = \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255.\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255.\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255.\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255.foo.example.com.
```

or, in alternate notation:

```
     \255{45}.\255{63}.\255{63}.\255{63}.foo.example.com.
```

Example where DNS name contains an octet which must be decremented by skipping values corresponding to US-ASCII upper-case letters:

```
x  = fo\[.example.com.

x' = \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255.\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255.\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255.fo\@\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255.example.com.
```

or, in alternate notation:

```
     \255{49}.\255{63}.\255{63}.fo\@\255{60}.example.com.
```

where {n} represents the number of repetitions of an octet.

Example where DNS name is the owner name of the apex, and consequently wraps to the DNS name with the maximum possible sort order in the zone:

```
x  = example.com.

x' = \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255.\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255.\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255.\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255.example.com.
```

or, in alternate notation:

\255{49}.\255{63}.\255{63}.\255{63}.example.com.

5.2 Examples of Immediate Successors

Example of typical case:

y = foo.example.com.

y' = \000.foo.example.com.

Example where DNS name is one octet short of the maximum DNS name length:

```
y  = fooooooooooooooooooooooooooooooooooooooooooooooooo\
     .ooooooooooooooooooooooooooooooooooooooooooooooooo\
     oooooooooooooooo.ooooooooooooooooooooooooooooooooo\
     ooooooooooooooooooooooooooooooooo.oooooooooooooooo\
     ooooooooooooooooooooooooooooooooooooooooooooooooo.example.com.
```

or, in alternate notation:

```
     fo{47}.o{63}.o{63}.o{63}.example.com.
```

```
y' = fooooooooooooooooooooooooooooooooooooooooooooooooo\
     \000.ooooooooooooooooooooooooooooooooooooooooooooooo\
     ooooooooooooooooooo.ooooooooooooooooooooooooooooooo\
     ooooooooooooooooooooooooooooooooooo.ooooooooooo\
     ooooooooooooooooooooooooooooooooooooooooooooooooo\
     oooo.example.com.
```

or, in alternate notation:

```
fo{47}\000.o{63}.o{63}.o{63}.example.com.
```

Example where DNS name is the maximum DNS name length:

```
y  = fooooooooooooooooooooooooooooooooooooooooooooooooo\
     o.ooooooooooooooooooooooooooooooooooooooooooooooooo\
     ooooooooooooooooo.ooooooooooooooooooooooooooooooooo\
     ooooooooooooooooooooooooooooooooo.oooooooooooooooo\
     ooooooooooooooooooooooooooooooooooooooooooooooooo\
     o.example.com.
```

or, in alternate notation:

```
fo{48}.o{63}.o{63}.o{63}.example.com.
```

```
y' = fooooooooooooooooooooooooooooooooooooooooooooooooo\
     p.ooooooooooooooooooooooooooooooooooooooooooooooooo\
     ooooooooooooooooo.ooooooooooooooooooooooooooooooooo\
     ooooooooooooooooooooooooooooooooo.oooooooooooooooo\
     ooooooooooooooooooooooooooooooooooooooooooooooooo\
     o.example.com.
```

or, in alternate notation:

```
fo{47}p.o{63}.o{63}.o{63}.example.com.
```


Example where DNS name is the maximum DNS name length and the least significant (left-most) label has the maximum sort value:

```
y  = \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\255\
     \255.ooooooooooooooooooooooooooooooooooooooooooo\
     ooooooooooooooooooo.ooooooooooooooooooooooooooooo\
     ooooooooooooooooooooooooooooooooooo.ooooooooooo\
     ooooooooooooooooooooooooooooooooooooooooooooooo\
     oooo.example.com.
```

or, in alternate notation:

```
     \255{49}.o{63}.o{63}.o{63}.example.com.

y' = ooooooooooooooooooooooooooooooooooooooooooooooooo\
     ooooooooooooop.ooooooooooooooooooooooooooooooooo\
     oooooooooooooooooooooooooooooo.ooooooooooooooooo\
     ooooooooooooooooooooooooooooooooooooooooooooooooo.\
     example.com.
```

or, in alternate notation:

```
     o{62}p.o{63}.o{63}.example.com.
```


Example where DNS name is the maximum DNS name length and the eight least significant (right-most) octets of the least significant (left-most) label have the maximum sort value:

```
y  = fooooooooooooooooooooooooooooooooooooooooo\255\
     \255\255\255\255\255\255\255.ooooooooooooooooooo\
     ooooooooooooooooooooooooooooooooooooooooooooo.ooo\
     ooooooooooooooooooooooooooooooooooooooooooooooooo\
```

```
     ooooooooooo.oooooooooooooooooooooooooooooooo\
     oooooooooooooooooooooooooo.example.com.
```

or, in alternate notation:

```
     fo{40}\255{8}.o{63}.o{63}.o{63}.example.com.
```

```
y' = fooooooooooooooooooooooooooooooooooooop.oooooo\
     ooooooooooooooooooooooooooooooooooooooooooooo\
     ooooooooo.ooooooooooooooooooooooooooooooooooo\
     ooooooooooooooooooooooo.oooooooooooooooooooo\
     ooooooooooooooooooooooooooooooooooooo.example.com.
```

or, in alternate notation:

```
     fo{39}p.o{63}.o{63}.o{63}.example.com.
```

Example where DNS name is the maximum DNS name length and contains an octet which must be incremented by skipping values corresponding to US-ASCII upper-case letters:

```
y  = fooooooooooooooooooooooooooooooooooooooooooooo\
     \@.ooooooooooooooooooooooooooooooooooooooooooo\
     ooooooooooooooooo.ooooooooooooooooooooooooooo\
     oooooooooooooooooooooooooooooooo.ooooooooooooo\
     ooooooooooooooooooooooooooooooooooooooooooooo\
     oo.example.com.
```

or, in alternate notation:

```
     fo{47}\@.o{63}.o{63}.o{63}.example.com.
```

```
y' = fooooooooooooooooooooooooooooooooooooooooooooo\
     \[.ooooooooooooooooooooooooooooooooooooooooooo\
     ooooooooooooooooo.ooooooooooooooooooooooooooo\
     oooooooooooooooooooooooooooooooo.ooooooooooooo\
     ooooooooooooooooooooooooooooooooooooooooooooo\
     oo.example.com.
```

or, in alternate notation:


```
     fo{47}\[.o{63}.o{63}.o{63}.example.com.
```

Example where DNS name has the maximum possible sort order in the zone, and consequently wraps to the owner name of the apex:

```
y  = \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255.\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255.\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255.\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255\255\255\255\255\255\255\255\255\255\255\
     \255.example.com.
```

or, in alternate notation:

```
     \255{49}.\255{63}.\255{63}.\255{63}.example.com.
```

```
y' = example.com.
```

6. Security Considerations

The derivation of some predecessors/successors requires the testing of more conditions than others. Consequently the effectiveness of a denial-of-service attack may be enhanced by sending queries that require more conditions to be tested.

7. IANA Considerations

This document does not create any IANA considerations.

8. Acknowledgments

The authors would like to thank Olaf Kolkman and Niall O'Reilly for

their review and input.

9. References

9.1 Normative References

[I-D.ietf-dnsext-dnssec-records]
Arends, R., Austein, R., Larson, M., Massey, D. and S. Rose, "Resource Records for the DNS Security Extensions", draft-ietf-dnsext-dnssec-records-11 (work in progress), October 2004.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.

9.2 Informative References

[I-D.ietf-dnsext-dnssec-trans]
Arends, R., Koch, P. and J. Schlyter, "Evaluating DNSSEC Transition Mechanisms", draft-ietf-dnsext-dnssec-trans-01 (work in progress), October 2004.

[I-D.weiler-dnsext-dnssec-online-signing]
Weiler, S. and J. Ihren, "Minimally Covering NSEC Records and DNSSEC On-line Signing", draft-weiler-dnsext-dnssec-online-signing-00 (work in progress), October 2004.

Authors' Addresses

Geoffrey Sisson
Nominet
Sandford Gate
Sandy Lane West
Oxford
OX4 6LB
GB

Phone: +44 1865 332339

EMail: geoff@nominet.org.uk

Ben Laurie
Nominet
17 Perryn Road
London
W3 7LR
GB

Phone: +44 20 8735 0686
EMail: ben@algroup.co.uk

Appendix A. Change History

A.1 Changes from -00 to -01

- Split step 3 of derivation of DNS name predecessor into two distinct steps for clarity.
- Added clarifying text and examples related to the requirement to avoid upper-case characters when decrementing or incrementing octets.
- Added optimisation using restriction of effective maximum DNS name length.
- Changed examples to use decimal rather than octal notation as per [RFC1035].
- Corrected DNS name length of some examples.
- Added reference to weiler-dnsext-dnssec-online-signing.
- Miscellaneous minor changes to text.

Intellectual Property Statement

Disclaimer of Validity

INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Acknowledgment

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