ENUM -- Telephone Number Mapping
Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 14, 2007

O. Lendl
enum.at

June 12, 2007

The ENUM Branch Location Record
draft-ietf-enum-branch-location-record-03

Status of this Memo

Copyright Notice

Abstract

This documents defines an extension to the E.164 Number Mapping (ENUM) algorithm by adding a mapping step which indicates where the ENUM tree for a specific ENUM application is located. A new DNS record (IEBL, the Infrastructure ENUM Branch Location record) is defined which provides an interim solution for the Infrastructure ENUM tree location.

Table of Contents

1. Introduction

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 [4].

ENUM (E.164 Number Mapping) as defined in RFC 3761 [1] (User-ENUM) is based on the concept of a single "golden" tree (e164.arpa) which stores telephone number to URI mappings.

Experience has shown that this single tree is not suitable for all applications and usage scenarios. The rules regarding administrative control of domains, opt-in requirements, and delegation hierarchy can vary between applications. See e.g. Infrastructure ENUM [6].

While non-terminal NAPTRs (see [3]) can redirect the ENUM resolution algorithm to another DNS tree, their semantics are not powerful enough to support an integration of Infrastructure ENUM into User ENUM at the number level.

A more generic application-specific redirection mechanism is thus needed.

An ENUM Branch Location Record as defined by this document contains information to drive a generalized algorithm which transforms a telephone number into a domain name. This extends the original algorithm as defined in section 2.4 of RFC 3761 [1] for specific use-cases.

This document defines the layout of a generic ENUM Branch Location (EBL) DNS Resource Record type and allocates a specific RRTYPE code for the Infrastructure ENUM use-case. In order for multiple such application to co-exist, each will need to allocate its own RRTYPE code.

2. Context

RFC 3761 defines ENUM as a Dynamic Delegation Discovery System (DDDS) application according to RFC 3401 ff [2]. As such, ENUM defines the following components of the DDDS algorithm:

1. Application Unique String
2. First Well Known Rule
3. Expected Output
4. Valid Databases

The generalized ENUM application extends only the definition of the

"Valid Databases" part of the DDDS algorithm. All other aspects of ENUM (e.g. further processing, valid enum-service types) are not affected.

The terminology can be confusing: ENUM is a DDDS Application. This draft generalizes ENUM to allow specific applications (e.g. Infrastructure ENUM) to use EBL records to tailor the ENUM algorithm to their individual needs. To distinguish these two layers of "applications", this document uses the term "use-case" for specific applications of the EBL-enabled ENUM algorithm.

This document does not define where EBL records are located in the DNS, that is left to documents which describe an actual use-case of the generalized ENUM application. These use-cases need to include a clear specification on where to look for EBLs, as well as allocate a RRTYPE code for this use-case.

3. The generalized ENUM Application

To recap, RFC 3761 (section 2.4) uses the following four steps as the "Valid Databases" part of the DDDS Algorithm:

1. Remove all characters with the exception of the digits.
2. Put dots (".") between each digit.
3. Reverse the order of the digits.
4. Append the string ".e164.arpa" to the end.

This small algorithm translates the "Application Unique String" (AUS, the E.164 telephone number) to a fully qualified domain name (FQDN) which is then used to query for NAPTR (Naming Authority Pointer, [3]) records containing rewriting rules.

Any use-case which uses EBL records to generalize the basic ENUM algorithm MUST clearly define where EBLs for this use-case are located in the DNS and MUST define the client behavior for the case if the EBL is not found at that location in the DNS tree. The EBL itself contains three parameters which affect the translation algorithm: SEPERATOR, POSITION, and APEX.

The generalized algorithm to derive the initial FQDN for the NAPTR lookup (thus replacing steps 1-4 from above) is defined as:

1. Apply the use-case specific algorithm to translate the AUS (the E.164 telephone number) to the location of the EBL record in the DNS. This needs to yield a fully qualified domain name (FQDN).

2. Query the DNS for an EBL record at the location of this FQDN, and retrieve the triple (SEPERATOR, POSITION, APEX) from this record.

   If multiple records are present, take any one and ignore the others.

   If no EBL record was found, the client MUST proceed according to the definition of the use-case, which could either be falling back to a default (e.g. use the triple ("", 0, "e164.arpa") to indicate the RFC 3671 "golden tree") or returning an error.

3. Build an ordered list of single-digit strings from all digits appearing in the AUS. All non-digit characters are ignored.

4. If SEPERATOR is not the empty string, then insert a string consisting of SEPERATOR after POSITION strings into this list. If the list of strings was shorter than POSITION elements, then report an error.

5. Reverse the order of the list.

6. Append a string containing APEX to the end of the list.

7. Create a single domain-name by joining the list together with dots (".") between each string.

Further processing is done according to RFC 3271: This domain-name is used to request NAPTR records which may contain the end result or, if the flags field is blank, produce new keys in the form of domain-names from the DNS.

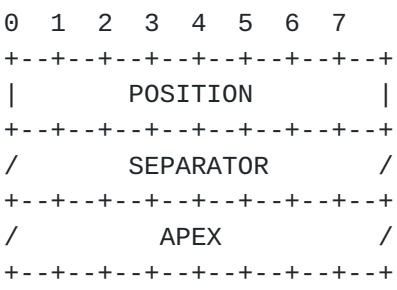
Section 5 contains examples.

4. The EBL Resource Record

Multiple use-cases of this algorithm can look for EBL records at the same location in the DNS. To distinguish EBL records from different use-cases, each use-case MUST allocate its own RRTYPE code for the EBL records associated with it. This document describes the generic RDATA format for all these EBL records, but allocates only the RRTYPE code for the Infrastructure ENUM use-case.

4.1. The EBL RDATA Format

The RDATA for an EBL RR consists of a position number, separator string and an apex domain:



where POSITION is a single byte, SEPARATOR is a <character-string> and APEX is a <domain-name>.

<character-string> and <domain-name> are defined in RFC 1035 [5].

The APEX field MUST NOT be empty; name-compression MUST NOT be used.

4.2. The EBL Presentation Format

The master file format follows the standard rules in RFC 1035. POSITION is represented as decimal integer. SEPARATOR is a quoted string, APEX is a domain name and thus does not require quoting.

4.3. The IEBL Record

The EBL record for the Combined Infrastructure ENUM use-case [7] is using the mnemonic "IEBL".

The RR type code for the IEBL RR is /IANA-ACTION/.

IEBL records are stored in the User-ENUM tree (e164.arpa) at the country-code (or group-of-countries) level, e.g. 1.e164.arpa, 3.4.e164.arpa, or 3.5.3.e164.arpa. A simple algorithm to determine the country-code length is given in draft-ietf-enum-combined-01 [7]. For up-to-date information regarding currently assigned country-code the see E.164 [8] and the ITU website under "ITU-T / Service Publications".

If no IEBL record is found at the country-code level then the ENUM client MUST report an error.

5. Examples

5.1. Combined Infrastructure ENUM

This example shows the use of IEBL records for the combined Infrastructure ENUM use-case.

This use-case defines that the IEBL resides at <reverse-country-code>.e164.arpa. Thus for example:

```
1.e164.arpa.      IN IEBL 4 "i" e164.arpa.
4.4.e164.arpa.    IN IEBL 2 "i" e164.arpa.
```

These records indicate how the transformation from E.164 number to ENUM domains for the use-case "Infrastructure ENUM" should be done for numbers in country-codes +44 and +1. This leads to the following mappings:

```
+1 21255501234          4.3.2.1.0.5.5.5.i.2.1.2.1.e164.arpa
+44 2079460123          3.2.1.0.6.4.9.7.0.2.i.4.4.e164.arpa
```

Here is the list of the intermediate steps for the second example to visualize how the algorithm as defined in Section 3 operates on "+44 2079460123":

1. According to the combined I-ENUM specification, retrieve the country-code from the number and build a FQDN using the reversed, dot-separated country-code and "e164.arpa", yielding "4.4.e164.arpa".

2. The IEBL lookup for this domain sets SEPERATOR to "i", POSITION to "2" and APEX to "e164.arpa".

3. The list of strings is ("4","4","2","0","7","9","4","6","0","1","2","3").

4. The SEPERATOR is "i", POSITION is 2, thus "i" is inserted between the second and the third string, yielding: ("4","4","i","2","0","7","9","4","6","0","1","2","3")

5. Reversing the list gives: ("3","2","1","0","6","4","9","7","0","2","i","4","4")

6. Appending APEX yields: ("3","2","1","0","6","4","9","7","0","2","i","4","4","e164.arpa")

7. Concatenation with dots: "3.2.1.0.6.4.9.7.0.2.i.4.4.e164.arpa"

After the introduction of the long term Infrastructure ENUM solution using "ienum.example.net" as the new apex for I-ENUM, the administrators of +44 can implement a smooth transition by changing its IEBL record in the following way:

```
4.4.e164.arpa.    IN IEBL 0 "" ienum.example.net.
```

This way, clients using the interim I-ENUM solution end up querying the same tree as clients implementing the long-term solution.

5.2. Tree Aggregation

EBL records can also be helpful in private ENUM settings. Consider a Voice over IP (VoIP) operator called "example.com" which participates in various country-specific VoIP peering services that all use their own private ENUM tree.

In order to avoid hardcoding country-specific ENUM lookups in its soft-switch, "example.com" can establish its own private ENUM tree which is populated with EBL records pointing to each fabric's ENUM tree. For example:

```
$ORIGIN enum.example.com.
1      IN IEBL 0 "" nanp-exchange.example.org.
4.4    IN IEBL 0 "" uk-peering.example.net.
3.5.3  IN IEBL 0 "" ie-link.example.net.
2.5.3  IN IEBL 0 "" enum.benelux.example.net.
1.3    IN IEBL 0 "" enum.benelux.example.net.
2.3    IN IEBL 0 "" enum.benelux.example.net.
```

6. Security Considerations

EBLs are used to direct ENUM resolvers to other places in the DNS. The security of DNS in both the location of the EBLs and wherever they point to needs to be maintained.

Use-case specifications need to be careful when designing their EBL location: Information concerning which numbers have been dialed could be leaked to the nameserver hosting the EBL records.

7. IANA Considerations

This documents allocates the Resource Records Type field for the IEBL record according to the definition in Section 4.

8. Acknowledgements

The author would like to thank Alexander Mayrhofer, Michael Haberler, Richard Stastny, Ed Lewis, and Olafur Gudmundsson for their contributions.

9. References

9.1. Normative References

[1] Faltstrom, P. and M. Mealling, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 3761, April 2004.

[2] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part One: The Comprehensive DDDS", RFC 3401, October 2002.

[3] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database", RFC 3403, October 2002.

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

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

9.2. Informative References

[6] Lind, S. and P. Pfautz, "Infrastrucure ENUM Requirements", draft-ietf-enum-infrastructure-enum-reqs-02 (work in progress), April 2006.

[7] Haberler, M. and R. Stastny, "Combined User and Infrastructure ENUM in the e164.arpa tree", draft-ietf-enum-combined-01 (work in progress), October 2006.

[8] International Telecommunications Union, "The International Public Telecommunication Numbering Plan", ITU-T Recommendation E.164, 1991.

Author's Address

Otmar Lendl
enum.at GmbH
Karlsplatz 1/9
Wien A-1010
Austria

Phone: +43 1 5056416 33
Email: otmar.lendl@enum.at
URI: http://www.enum.at/

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).