ENUM -- Telephone Number Mapping
Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 15, 2007

O. Lendl
enum.at

December 12, 2006

The ENUM Branch Location Record
draft-ietf-enum-branch-location-record-02

Status of this Memo

Copyright Notice

Abstract

This documents defines the ENUM Branch Location record (EBL) which is used to indicate where the ENUM tree for special ENUM application is located. The primary application for the EBL record is to provide a temporary solution for the Infrastructure ENUM tree location.

Table of Contents

1. Introduction

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 [5].

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.

The ENUM Branch Location (EBL) 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.

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 as based on EBL records 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.

3. The generalized ENUM Application

To recap, RFC 3761 (section 2.4) uses the following steps for 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 needs to define where EBLs for this use-case are located in the DNS. 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 yields 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, use the triple ("", 0, "e164.arpa") as default. This corresponds to the RFC 3671 "golden tree".

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

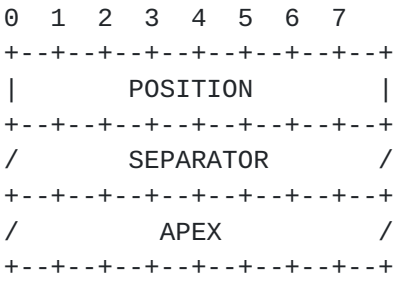
Section 5 contains examples.

4. The EBL Resource Record

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

4.1. The EBL RDATA Format

The RDATA for a EBL RR consists of a position number, separator string and an apex domain. <character-string> and <domain-name> refer to the definitions of RFC 1035 [4].



where POSITION is a single byte, SEPARATOR is a <character-string> and APEX is a <domain-name> and must not be empty. Name-compression is not to be used for the APEX field.

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.

5. Examples

This example shows the use of EBL records as defined by the interim

solution to Infrastructure ENUM as defined by draft-ietf-enum-combined-01 [6].

This application defines that the EBL resides at "infrastructure".<reverse-country-code>.e164.arpa. Thus for example:

```
infrastructure.3.4.e164.arpa.    IN EBL 2 "i" e164.arpa.
infrastructure.1.e164.arpa.      IN EBL 4 "i" example.com.
infrastructure.9.4.e164.arpa.    IN EBL 0 ""  ie164.arpa.
```

These records indicate how the transformation from E.164 number to ENUM domains for the application "Infrastructure ENUM" should be done for numbers in country code +43, +1, and +49. This leads to the following mappings:

```
+43 15056416           6.1.4.6.5.0.5.1.i.3.4.e164.arpa
+1 5551234567          7.6.5.4.3.2.1.i.5.5.5.1.example.com
+49 891234567          7.6.5.4.3.2.1.9.8.9.4.ie164.arpa
```

Here is the list of the intermediate steps for the first example to visualize how the algorithm as defined in Section 3 operates on "+43 15056416":

1. According to the interim, combined I-ENUM specification, retrieve the country-code from the number and build a FQDN using "infrastructure", the reversed, dot-separated country-code and "e164.arpa", yielding "infrastructure.3.4.e164.arpa".

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

3. The list of strings is ("4","3","1","5","0","5","6","4","1","6").

4. The SEPERATOR is "i", POSITION is 2, thus "i" is inserted between the second and the third string, yielding:
   ("4","3","i","1","5","0","5","6","4","1","6")

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

6. Appending APEX yields:
   ("6","3","4","6","5","0","5","1","i","3","4","e164.arpa")

7. Concatenation with dots: "6.3.4.6.5.0.5.1.i.3.4.e164.arpa"

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 EBL record according to the definition in Section 4.

8. Acknowledgements

The author would like to thank Alexander Mayrhofer, Michael Haberler, Richard Stastny, Klaus Nieminen, Richard Shockey, and Karsten Fleischhauer 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] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

9.2. Informative References

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

[6] 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.

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).