Experiences from implementing ENUM system

IETF59 Seoul enum wg

March 3, 2004

Kazunori Fujiwara <fujiwara@jprs.co.jp>

Experiences from implementing ENUM system

- What JPRS is implementing
  - ENUM Registry system
    - Web I/F
    - Registry DB (including zone file generator)
  - Visual ENUM DNS resolver
    - Windows application
    - Web CGI (Perl Module)
- What we found
  - Ambiguousness of RFC2916bis and DDDS RFCs
  - draft-conroy-enum-experiences-01 is useful
  - Clarification should be done by
    - updating RFC2916bis and DDDS RFCs, or
    - writing a BCP document from operational point of view

(1) Regular Expression Interpretation

- Ambiguousness of NAPTR RR regular expression interpretation.
- RFC2916 3.2.3 Example3:

  * IN NAPTR 100 10 “u” “ldap+E2U”
  "!^+46(.*)$!ldap://ldap.se/cn=\1!" .

  – '+' is a meta character in POSIX regular expression, and this '+' is used as a literal character, '+' character must be escaped by '\'
- Which is correct?
  – !^+813(.*)$!sip:\1@domain! bad?
  – !^\+813(.*)$!sip:\1@domain! OK?

(2) Selection of Regexp delim-char

- RFC3402 says Rexexp field delim-char may be any octet not in 1-9 or i.
- In the real usage, several characters are troublesome.
- There are sample REGEXP fields. Is it correct?

  \^.*$\mailto:fujiwara@jprs.co.jp\

  \^+813(.*)$\sip:\1@domain\

  <\000>^.*$<\000>sip:some@domain<\000>
- MUST we accept \ or character code 0 or 0x80-0xff in ENUM client?
- I propose we should restrict delim-char to some special characters such as '!'.

(3) processing order ambiguoursness SERVICE before ORDER? (1/2)

- DDDS RFC says ORDER must be processed first.
- In RFC3403 page 9, there is a E164 Example.

```
$ORIGIN 2.1.2.1.5.5.5.0.7.7.1.e164.arpa.
    IN NAPTR 100 10 "u" "sip+E2U"   "!^.*$!sip:information@foo.se!i"     .
    IN NAPTR 102 10 "u" "smtp+E2U" "!^.*$!mailto:information@foo.se!i" .
```

- If we MUST process in ORDER-field order without checking SERVICE-field, we MUST choose order=100 entry only.
- RFC3404 Appendix A. Pseudo Code seems to be such code.
- But in RFC3403 same page,

"These state that the available protocols used to access that telephone's service are either the Session Initiation Protocol or SMTP mail."

- This seems contradiction.

(3) processing order ambiguoursness SERVICE before ORDER? (2/2)

- So, I assume that ENUM client match Service field first and then, process ORDER field.
- Already pointed out by draft-conroy-enum-experiences-01
  - section 2.9 first paragraph is very important.
  - It must be independent section.

(4)ENUM always return single rule?(1/2)

- rfc2916bis page 8, section 2.5 'Enum Resolver'
  - 'The ENUM algorithm always returns a single rule.'
  - but such sentence (a single rule) is first shown in here.
- I'm still confusing this paragraph.
  - I want to write multiple contact points in ENUM.
  - I think people who need to call me SHOULD try all ENUM entries.
  - Retry process may be done automatically.
- Examples
  - Two SIP URIs in same order
    - IN NAPTR 100 10 "u" "!^.*$!sip:fujiwara@jprs.co.jp!" .
    - IN NAPTR 100 20 "u" "!^.*$!sip:fujiwara@wide.ad.jp!" .
  - SIP and email URIs in same order
    - IN NAPTR 100 10 "u" "!^.*$!sip:fujiwara@jprs.co.jp!" .
    - IN NAPTR 100 20 "u" "!^.*$!mailto:fujiwara@jprs.co.jp!"

(4)ENUM always return single rule? (2)

- Then, I decided an ENUM client does not fallback automatically.

- I assume ENUM Resolver means ENUM client application.
- Our ENUM library returns multiple URIs which has same ORDER value.
- We implement that users may choose one URI from URI lists displayed by ENUM application.

(5) MUST process non-terminal NAPTR?

- DDDS non-terminal loop processing seems unclear.
  - Already pointed out by draft-conroy-enum-experiences-01
- DDDS non-terminal NAPTR processing is too complicated to implement ENUM function to small devices.
- non-terminal NAPTR processing may cause large delay.
- So, I decided that our implementation ignores non-terminal NAPTRs.

Summary: What we did

- Our implementation
  - ignores non-terminal NAPTRs.
    - to decrease delay.
  - returns multiple URIs.
    - User may select one of URIs.
  - process SERVICE match before ORDER field sorting.
  - accept '\' as delim-char.
    - but may not work when delim-char is ‘\000’ or 0x80-0xff.
  - can handle multiple services
    - my cellular phone can handle http:, mailto:, tel: URIs in JAPAN.
- Is this RFC2916bis compliant?
- Any suggestions, comments and advices are welcome :-)