Converting a CIDR Address to Reverse-DNS Name

Quick Overview of Naming Convention

IPv4 and IPv6

SECURE64

SOFTWARE CORPORATION

Converting IPv4 CIDR to reverse-DNS

- Invert the address per the usual reverse-DNS method. Remove any trailing zeroes.
  - » 129.82.0.0/16 --> 82.129.in-addr.arpa
- Calculate `N = prefix-length mod 8.`
- if N = 0, you are at an octet boundary and are done.
- Otherwise:
  - add an “m” character to indicate “mask”
  - convert the least significant octet to binary, separate with “.” characters
  - truncate to the “N” significant binary characters for this prefix length
  - reverse the string per reverse DNS
- Examples: (showing step 1: “convert to binary”, and step 2: “truncate and reverse”)
  - 129.82.64.0/18 --> 129.82.m.0.1.0.0.0.0.0.0 --> 1.0.m.82.129.in-addr.arpa.
  - 129.82.64.0/20 --> 129.82.m. 0.1.0.0.0.0.0.0 --> 0.0.1.0.m.82.129.in-addr.arpa.
  - 129.82.160.0/20 --> 129.82.m.1.0.1.0.0.0.0.0 --> 0.1.0.1.m.82.129.in-addr.arpa.
  - 129.82.160.0/23 --> 129.82.m.1.0.1.0.0.0.0.0 --> 0.0.0.0.1.0.1.m.82.129.in-addr.arpa.

Converting Reverse-DNS Name to CIDR

- Mask length = 8*octets + number of binary digits
- Reverse the string. Add up the values of the binary digits to calculate the final octet. Append the "/" and mask length.
    - 1.0.m.82.129.in-addr.arpa   --> 129.82.64.0/18
        - example has 2 octets + 2 binary digits, so mask length = 18
    - 0.0.1.0.m.82.129.in-addr.arpa --> 129.82.64.0/20
        - example has 2 octets + 4 binary digits, so mask length = 20
    - 0.0.0.1.0.1.m.129.in-addr.arpa --> 129.160.0/14
        - example has 1 octet + 6 binary digits, so mask length = 14

SECURE64

Converting IPv6 CIDR to reverse-DNS

- The same idea, just at nibble boundaries. So it is easier.
- Invert the address per the usual reverse-DNS method. Remove any trailing zeroes.
    - » 2607:fa88::/32 -->   8.8.a.f.7.0.6.2.ip6-arpa.
- Calculate `N = prefix-length mod 4.`
- if N = 0, you are at a nibble boundary and are done.
- Otherwise:
    - ➢ add an “m” character to indicate “mask”
    - ➢ convert the least significant nibble to binary, separate with “.” characters
    - ➢ truncate to the “N” significant binary characters for this prefix length
    - ➢ reverse the string per reverse DNS
- Examples: (showing step 1: “convert to binary”, and step 2: “truncate and reverse”)
    - ➢ 2607:fa88:8000/33   --> 2607:fa88.m.1.0.0.0 -->   1.m.8.8.a.f.7.0.6.2.ip6.arpa.
    - ➢ 2607:fa88:8000/34   --> 2607:fa88.m.1.0.0.0 -->   0.1.m.8.8.a.f.7.0.6.2.ip6.arpa.
    - ➢ 2607:fa88:c000/35   --> 2607:fa88.m.1.1.0.0 --> 0.1.1.m.8.8.a.f.7.0.6.2.ip6.arpa.

Next draft

- May decide to always append an “m”, even at octet or nibble boundary.

  - » 2607:fa88::/32 --> m.8.8.a.f.7.0.6.2.ip6-arpa.

SECURE64