Zone Clones

Duplicate Namespace Provision Using Zone Clones (CLONE RR)

DNSEXT Virtual Interrim Meeting
16-February-2010

Paul Vixie, ISC

Motivation

- {B,C,D}NAME cannot be the target of MX/NS/etc
- Thus name equivilences created in that way are not *first class* names – they can only be used at the application/presentation layer
- Registries/registrars can trivially insert parallel NS RR sets for namespace replication, but registrants and their nameserver operators will need new tools
- These new tools will need some DNS metadata to coordinate their operation

Constraints

- Must not require stubs or recursives to be upgraded, since there are millions of these and the tail is long
- Must be an Internet Standard, not a proprietary or adhoc extension, to facilitate multivendor operation
- Must not place any burden on registry, which may be regulated (so, autoinsertion into root zone, *no!*)
- Authority server operators, protocol implementors, and registrars can accept burdens, since they have incentives, and are few in number

Assumptions

- A registrant can select and direct its nameserver operators in light of new Internet Standards
- Implementors and operators can respond to market pressures or not, regarding new feature development
- A requirement that all of a zone's authority servers be upgraded before that zone can support a new feature like “zone clones” is thus reasonable
- Not all equivilanced namespaces will have the same parent, e.g., `vixie.sf.ca.us` vs. `vix.com`.

Principles

- Only the primary master knows the content of RDATA beyond the RR types from RFC1034/1035, they are opaque to secondary and recursive servers
- Replication by multi-apexing is thus possible only on a primary master – in BIND terms this is using the same zone file for multiple zones, where parsing has to know the zone apex for non-qualified names in any RDATA
- Therefore we're going to do a lot of IXFR/AXFR's in parallel, and use a lot of RAM on the secondaries

Details (1)

- New RR added to zone apex: CLONE
  - `$ORIGIN vix.com.`
    `@ IN CLONE vixie.com.`
    `@ IN CLONE vixie.sf.ca.us.`
- No special processing required by stub, forwarder, or recursive servers, nor by applications/consumers
- Primary server synthesizes the zones named as CLONE (for example, by loading the zone multiple times using different apexes to guide the parsing of unqualified names in RDATA's.)

Details (2)

- Secondary servers synthesize the zones named as CLONE and performs normal IXFR/AXFR to pull the content from upstream
- Deep IXFR/AXFR dependency graphs are supported as normal, the CLONE RR is just configuration-level “syntactic sugar”
- Care has to be taken regarding collisions, if a zone is named by multiple CLONE RR sets or by a CLONE RR set and also by a directly configured zone (open question: is this a hard failure or just a warning?)

Limitations

- Only works for leaf zones, since a delegation would have to propagate its CLONE RR to its children (open question: should we allow such propagation or allow grandchildren to search upward for CLONE? *Note: this could get very expensive!*)
- Requires strong trust between registry and registrar/registrant, this is probably a high-fee service that would not be enabled by default
- Requires key sharing among zone and its clones, and multiple zone signing events (for DNSSEC)

Summary

- New RR added to zone apex: CLONE
- Upgrade required for registrant's authority servers
- Creates first class names
- Creates second class namespaces (leaf-only)