

# *DNSSECbis Lookaside Validation*

Paul Vixie <[vixie@isc.org](mailto:vixie@isc.org)>  
Internet Systems Consortium  
(December 2005)

---

---

# *Topics*

- Introduction
  - DNS Delegation and Resolution
  - DNSSECbis Data and Traversal
  - DLV Overview
  - DLV Operations
  - Conclusion
- 
-

# *Introduction*

- DNS in 1987 was to replace HOSTS.TXT and allow for future expansion
  - Authenticity of DNS data (or anything else on the Internet, for that matter) wasn't considered
  - From 1994 to 2005 (and beyond), IETF designed and redesigned Secure DNS
  - Secure DNS deployment depends on miracles
  - DLV is a (subversive?) early deployment aid
- 
-

# DNS Data and Delegation

- Domain names lay inside a hierarchy of *zones*
    - every zone except “the root” has ancestors
    - any zone can have descendants, by *delegation*
    - “root” zone is ultimate ancestor of all zones
    - every zone has some *authority* servers
  - DNS nodes can contain *resource record sets*
    - sets denoted by  $\langle name, type, class \rangle$  (A, MX, NS, etc)
    - each record has some kind of data (IP or IP6 address, mailserver, nameserver, or whatever)
    - NS RR set introduces a child zone (*delegation point*)
- 
-

# *DNS Traversal and Recursion*

- A server is authoritative for zero or more zones
    - zero zones == caching forwarder
  - Authority response types
    - negative: “no name matches your qname”
    - empty: “name is good, but no rrsets match your qtype”
    - positive: “here's what you asked for”
    - referral: “that's in a subzone, go ask somebody else”
  - Caching forwarder behaviour
    - acts on behalf of “stub” resolvers
    - caches data for reuse, follows referrals, etc
    - configured to know list of “root” zone servers
- 
-

# DNSSECbis Data and Traversal

- New DNS metadata RR types
    - DNSKEY: public key, found at a zone's apex
    - RRSIG: generated using RR set data + private key
    - NSEC: authenticates unused name space
    - DS: in parent zone, authenticates zone's DNSKEY
  - Validation
    - Positive answers will include an RRSIG (+ DNSKEY)
    - Referral answers will contain a signed DS (with NS)
    - Negative or empty answers will contain an NSEC
    - Validator is configured to know some *trust anchor(s)*
      - ultimately this means knowing the public key for “root”
- 
-

# *Problems in DNSSECbis Approach*

- Trust anchors are *very* widely distributed
    - there's no way to roll out a new key more than once
    - therefore the number of useful anchors is likely “one”
    - and that “one” has to last for the Internet's lifetime
  - Root zone stewardship is politically complicated
    - signing the root zone requires a strong permanent key
    - DNSSECbis depends on trust among root's stewards
    - current stewards (ICANN, ++ ) are not mutually trustful
  - Economic benefits of DNSSECbis are unclear
    - adds value for DNS data consumers and producers
    - adds great cost, little revenue for registries/registrar
    - DNS autonomy means “monopoly powers” (.COM)
- 
-

# DLV Overview

- Local policy mechanism for validators
    - not an IETF standard – producer/consumer “co-op”
    - only affects results that would have been unsecured
  - Early deployment aid
    - supports market growth from 0%, but not full Internet
    - to be killed when “root” and some gTLDs are secured
  - Supports/expects migration to “real DNSSECbis”
    - lets producers/consumers have Secure DNS now
    - creates a market to support registry/registrar costs
    - allows politicians more time to improve stewardship (“hope springs eternal”)
- 
-

# DLV Metadata

- DLV resource record
    - structurally identical to DS RR (differs semantically)
    - RR type code number is from experimental space
  - DLV namespace
    - is within normal DNS namespace
    - normal DNSSECbis is used to secure it
    - can have normal interior zone cuts and delegations
  - Example
    - DLV namespace at DLV.ISC.ORG
    - DNSKEY exists for VIX.COM
    - no DS for VIX.COM (or, most likely, for COM)
    - insert DLV RR at VIX.COM.DLV.ISC.ORG
- 
-

# *DLV Validation*

- Validators are configured with one or more DLV namespaces and trust anchors
  - Whenever normal DNSSECbis metadata cannot be found or validated...
    - select the best matching DLV namespace known
    - select the best matching DLV RR within that space
  - Examples
    - if two DLV name spaces are known, “root” and MIL
      - no MIL name would ever be searched in the “root” DLV
    - if a DLV namespace knows COM and VIX.COM
      - the VIX.COM DLV would take precedence over COM's DLV for queries of VIX.COM, WWW.VIX.COM, etc
- 
-

# Aggressive Negative Caching

- Possibility of MiM attacks requires that validator issue many DLV queries
  - Cached NSEC RRs could obviate these queries
  - Problem: NSEC not intended for negative caching
  - Solution: *Off-The-Wire* negative caching
    - the DLV logic in the validator is “like an application”
    - applications are free to interpret cached NSECs
  - Result: most DLV queries will be suppressed
  - Example
    - cached NSEC declared nonexistence between AAA.DLV.ISC.ORG and CCC.DLV.ISC.ORG
    - no need to query for BBB.DLV.ISC.ORG
- 
-

# *DLV Operations*

- DLV Registry: accept public keys from verified zone owners over repudiable channels; publish
    - should be public benefit corporation with cost-based fee structure, who will kill off DLV when time comes
  - DLV Registrant: submit to DLV Registry the DNSKEY values from signed zones
    - submissions can cease once the zone's parent is secured, if parent uses DLV or if DLV is dead/dying
  - Validator Operators: retrieve and configure trust anchors and DLV namespace info from Registry
    - monitor registry in case of key rollover events
- 
-

# Conclusion

- Secure DNS is urgently and much needed by users
    - but has no viable economic or deployment model
  - DLV is an early deployment aid
    - should scale well enough
    - shouldn't scale too well
  - DLV is not an IETF standard – just a “co-op”
  - ISC is committed to DLV
    - will support DLV in BIND9 (9.4.0, due “soon”)
    - will operate a robust DLV registry (similar to f-root)
    - will kill DLV when the need for it passes
- 
-

# Questions

- Who else worked on this?
    - David Conrad, Johan Ihren, Mark Koster, Sam Weiler, Mark Andrews, and many others
    - Nobody endorses it other than Paul Vixie and ISC
  - Why isn't this an IETF protocol?
    - deployment is “just a detail” (ivory-towerism?)
  - Why did ISC decide to do DLV?
    - our mission statement made us do it
  - Is this work published anywhere?
    - Google for “ieice vixie dlv” to get the 2004 paper
  - What else?
- 
-