New Challenges and Dangers for the DNS

Jim Reid
ORIGIN TIS-INS

Jim.Reid@nl.origin-it.com

Introduction

- new technologies
  - IPv6, W2K
  - dynamic DNS updates
  - secure DNS
- new resource records
  - NXT, KEY, SIG, TSIG
  - AAAA, SRV, IXFR
  - A6, DNAME
- close inter-relationships
- probably unavoidable

IPv6

- **128-bit addresses**
  - **cumbersome**
- **reverse lookups**
  - `ip6.int` **domain**
  - **analagous to** `in-addr.arpa`

W2K

- WINS dies (rejoice!)
  - — replaced by Active Directory Service
  - — depends on SRV records
- dynamic DNS updates
  - — WINS-like on the fly registration:
    - names and addresses
    - services - printing, dialup, etc

Secure DNS

- strong authentication
  - name servers
  - queries
- industrial-strength crypto
  - Diffie-Hellman, RSA
- strong checksumming
  - DSS, MD5

Dynamic Updates

- **on the fly updates of zone data**
- **needed for plug & play**
- **updates SOA zone version number**
- **BIG problems**
  - **— security**
    - **write access to zone data**
  - **— scaling**
    - **zone transfer storms on Monday morning**
  - **— zone synchronisation**
    - **who updates forward & reverse zones?**

Dynamic DNS Scaling Worries

- each (set of) updates bumps SOA
  - => zone transfer to slaves
- get DHCP server to batch updates?
- writing transaction logs on name servers will slow this anyway

Dynamic DNS - Security Worries

- **who gets write access to DNS zone?**
- **no fine-grained control**
  - **anyone can change just about anything**
  - **obviously not for desktops**
- **only for "trusted" systems**
  - **sane DHCP servers**
  - **even then use secure Dynamic DNS**

Dynamic DNS & W2K

- **W2K depends on Dynamic DNS**

- **makes DNS more WINS-like**

- **who wants a W2K box scribbling on their DNS data?**

  — **put 'em in a leper colony**

  — **delegate** `w2k.foo.bar` **(say)**

  — **make** `ntbox.foo.bar` **a CNAME for** `ntbox.w2k.foo.bar`

- **weird WINS-like names**

  — **undocumented**

    - **JSPNRMPTGSBSSDIR from Remote Access Service**

Dynamic DNS - Forward and Reverse Zones

- who does what?
- DHCP server does forward and reverse updates
  - "atomic" operation
  - least insecure method
  - dynamic name/address mappings
    - not good for dial-in pools
  - what about fixed names or addresses?
    - bind names and IP addresses to MAC addresses?
    - might need this for IPv6

Dynamic DNS - Forward and Reverse Zones

- **DHCP server does forward update, client updates reverse zone**
  - **seems to be the W2K approach**
  - **asynchronous forward/reverse updates**
  - **dial-ins can assign fixed names**
  - **do you want random computers updating the DNS?**
    - **scaling and security worries again**

Secure Dynamic Updates

- RFC2137
- crypto authentication
- a bit of a misnomer
  - only authenticates the request
  - no say over what the request changes

DHCP & Dynamic DNS

- not much happening
  - ISC DHCP development stalled
  - Microsoft could well drive this
- use static names in DNS (for now)
- hassles for roaming users
  - move away from host-based authentication in long run?

Incremental Zone Transfer

- **RFC1995**
- **IXFR query type**
- **send deltas, not whole zones**
- **meant for** `.com`
- **implemented in BIND8.2**
  - **special case of dynamic updates**
  - **comparable semantics**

New Resource Records

- SRV
  - service location
- SIG
  - crypto-signature for a RR
- NXT
  - what RRs have SIG records
- KEY
  - public keys of SIG records
  - shared secrets for TSIG?

More New Resource Records

- **AAAA**
  - **IPv6 addresses**
- **A6**
  - **map a domain name to an IPv6 address**
  - **IPv6 delegation & reverse lookup**
  - **should replace AAAA**
- **DNAME**
  - **CNAMEs for domains**

The SRV RR

- **RFC2052**
  - **due for update Real Soon Now**
- **format:**

_Service._Proto.Name SRV Priority Weight Port Target

- **example:**

`_http._tcp.www.a.net. SRV 0 0 80 foo.bar.`

  - **web service for** `www.a.net` **is on TCP port 80 of foo.bar.**
  - **priority field is like MX priority**
  - **weight field is for crude load balancing**
  - **underscores in new standard**

The TSIG RR type

- on standards track, no RFC yet
- transaction signatures
- lightweight authentication
- relies on a shared secret:
  - HMAC-MD5
  - other algorithms possible
- not in zone files
  - computed on the fly
  - appended to additional data section

The KEY RR

- **defined in RFC2065**
- **public key for some name**

format:

```
name KEY flags proto algorithm public-key
```

- **flags - what kind of key?**
  - **user, zone, IPsec, etc**
- **proto - identify non-DNS applications**
  - **SSH?, SSL?, email, IPsec, Kerberos? keys**
- **crypto algorithm - MD5/RSA**
- **base-64 encoding of key**

Example KEY RR

```
foo.com. IN KEY 513 3 1 ( \
    AQOxuZdEyFDlONGz9xF3fdAvG \
    PaUqj6s727UOXVtXKcyodC0EM \
    C+82L1cDFa1AqsgPrMjHRqfzL \
    iaAoVKYPof+sdWr+fD/DGzKAx \
    nK1FKRMRTyDoZnk3uqfje5n2Q \
    uSDDMZPKhEt1qwISzowjJZCGU \
    WU1wyH/B7TPTvuaPen/ExayQ== \
)
```

The SIG RR

- **also defined in RFC2065**

format:

```
name SIG type flags proto algorithm \
time-RR-signed sig-expiry-time \
footprint signer signature
```

- **type is the RR type that is signed**
- **proto, flags and algorithm identify crypto**
- **timestamps thwart cryptanalytic replay and replay attacks**
  - **=> secure NTP**
- **signer: who signed the SIG**
- **signature in base-64 encoding**

The SIG RR continued

- each SIG RR signs 1 resource record
- signer identifies relevant KEY RR
- delegated signing authority
  — postmaster could sign MX records

Example SIG RR

```
bar.foo.com. SIG MX 1 3 ( \
    19960102030405 \
    19961211100908 \
    21435 \
    foo.com. \
    MxFcby9k/yvedMfQgKzhH5er0Mu/ \
    vILz45IkskceFGgiWCn/GxHhai6V \
    AuHAoNUz4YoU 1tVfSCSqQYn6//1 \
    1U6Nld80jEeC8aTrO+KKmCaY=
)
```

The NXT RR

- **defined in RFC2065**
  - **— which RRs are signed or not**
  - **— authentication of non-existent names**
- **RR type not found in zone files**
  - **— derived from zone contents**
  - **— in auth. section of reply from a secure name server**
- **example:**

```
foo.bar.com. NXT foo.bar.com. A NXT
```

SIG/KEY RR Generation

- **primitive tools in BIND8.2**
  - **dnskeygen**
  - **dnssigner**
- **scant documentation**

Interesting SIG/KEY/TSIG Problems

- **signing zone transfers**
- **wildcard resource records**
- **normalised RR names:**
  - **all lower-case**
  - **fully qualified domain names**
  - **standard TTL values**
  - **what original data was signed?**

Key Management

- **a very hard problem**
  - **— but we already knew that...**
- **private keys and shared secrets in** `/etc/named.conf`
  - **—** `server` **statements**
  - **—** `key` **statements**
  - **— very ugly**
  - **— an N-squared problem**

Secure DNS Problems

- public-key crypto is expensive
  - — not for common usage
  - — signing "important" data
    - zone transfers?
    - keys, e-commerce?
- TSIG is computationally cheap-ish
  - — maybe for resolving?
    - shared secret a problem: can’t be secret
  - — probably OK dynamic DNS
    - "trusted" DHCP servers

Secure DNS Concerns

- establishing relationships of trust between name servers
  - master and slave servers
  - intra- and inter-domain
    - does foo.com. "trust" com.?
    - does foo.com. "trust" bar.com.?
    - does com. "trust" foo.com.?

Secure DNS and Top Level Domains

- query rate on TLD name servers:
  - — ˜2000/sec on Internet root server
  - — where is the compute power for even TSIG?
- key management for .com domain
  - — ?million `key` & `server` statements?
- memory usage
  - — signing every RR makes zone 10x bigger!
  - — currently ˜600 Mb for unsigned .com domain

The AAAA RR

- **defined in RFC1886**
- **IPv6 notation from RFC1884**

```
example IN AAAA 1080:0:0:0:8:800:200C:417A
example IN AAAA 1080::8:800:200C:417A
example IN AAAA 1080:0:0:0:8:800:32.12.65.122
example IN AAAA 1080::8:800:32.12.65.122
```

- **unwieldy PTR records**

```
b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0. \
0.1.0.0.0.0.0.0.0.1.2.3.4.IP6.INT. PTR example
```

- **may be obsoleted by A6 RR type**

The A6 RR

- no RFC yet - on standards track
- two or three fields
  - prefix length
  - textual representation of IPv6 address
  - domain name if non-zero prefix length
- example

```
CC.NET.ALPHA-TLA.ORG. A6 0 2345:00C0::
```

  - C.NET.ALPHA-TLA.ORG "owns" IPv6 addresses beginning 2345:00C0

The DNAME RR

- **no RFC yet - on standards track**
  - **format:**
    owner DNAME target
  - **example:**
    `d.e.f. DNAME w.xy.`
    - **lookup of a.b.c.d.e.f => lookup of a.b.c.w.xy**
- **useful with:**
  - **A6 records**
  - **RFC2317-style delegations**

IPv6 and DNAME/A6 Records

- A6 & DNAME records are cleaner
  - smaller and simpler `ip6.int` zone
  - easier to manage & delegate
    - regional, provider, subscriber bits
  - parallel address spaces
    - easier renumbering!
- should replace AAAA records
- bottom bits come from MAC address
  - => dynamic DNS?