PowerDNS & General Thoughts on the (Ir)relevance of DNS

bert hubert
ahu@ds9a.nl
Netherlabs Computer Consulting BV

Outline of this presentation

What is PowerDNS

- What did we do differently
  - Our attitude
  - C++/Threads
- What do we suck at
- Performance
- The Recursive Recursor
- What did we learn from our experiences
- General thoughts on the (ir)relevance of DNS

What is PowerDNS

- Open Source (GPL version 2) Authoritative Nameserver
- Open Source Recursive recursing Nameserver
- Full Master and Slave support
- Talks to ODBC, MySQL, PostgreSQL, LDAP, TDB/GDB/DB2, IBM DB2, Oracle, Pipes and BIND style zonefiles & configurations
- Available for Unix and Windows
- Full IPv6 support

What did we do differently

- Started out as a commercial company with a closed source proprietary product
- Heavy use of the C++ Standard Template Library
- Some threading, some 'fibers' (MTasker)
- Zone files are just one data source
- No support for every OS (HP/UX, SCO :-))
- No desire to exactly follow some of the sillier parts of the STD documents
- Whole packet caching

What do we suck at

- No real support for EDNS0
  - we blindly truncate at 512 octets
  - will happily accept larger answers though
- Support for large queries that need TCP in the recursor is shaky to say the least
- We could go single threaded for many databases
- Solaris support lags behind
  - we have no Sun
- Packet parser is primordial code
  - PowerDNS was my first C++ program!
- We store IPv6 and IPv4 addresses as ASCII!

Our attitude

- We want stuff to Work Well, no DJB-isms.
- Don't send out ritualistic bogus data however
- Security over everything
- Be a friendly netizen
  - don't flood remote nameservers with queries
  - give answers that can be parsed well
  - robustness principle, adhere to relevant specs
- We do not follow newfangled DNS developments unless we see real use or demand among our users
- Trade utmost efficiency over straightforward code
  - for example, we store IP addresses as ASCII

Intermezzo: C++ & Threads Evangelism

- C++ has a bad rap
- Mostly an attitude issue – C++ generally not associated with the lean & mean crowd
- `typedef map<string,set<ResourceRecord> > cache_t;`
- Concurrent programming makes for sequential easy to read code

Performance

- We strive to offer the operator choice
- Keep threading to a minimum
- Availability of all features comes at a performance cost
- Things that can be switched off:
  - CNAMES
  - Wildcards
  - Out of zone and IPv6 additional processing
  - Logging
  - Strictly RFC compliant AXFRs (1 record/packet)
- Out of the box, performance is not the priority
- Ability to serve millions of zones (tested) is!

Specific performance features

- We don't check for a SOA unless we need to!
  - We assume a competent operator :-)
  - Query for DS9A.NL/MX results in 1 SQL query if it exists. This breaks RFC1034 Algorithm.
- Whole packet caching
  - An identical packet (except for the id) gets answered within a microsecond, id is spoofed copied in
- Database query caching, negative query caching
- No authority records unless needed

The recursor (1/2)

- Cooperative multitasking using MTasker
- 1200 lines of code
- Impressive array of features:
  - Verisign oddity removal
  - Query throttling
    - throttles lame results for nameserver/zone tuples
    - throttles SERVFAIL responding nameservers/zone tuples
    - throttles non-responding nameservers
  - Fastest nameserver selection
    - full RTT decay
- Completely separate from authoritive nameserver

The recursor (2/2)

- --trace output very useable to debug DNS problems
- Memory cache, persistent cache (in CVS)
- Fully recursive recursing nameserver
- In an adnslogres reverse lookup test from a cold cache, generally many times faster than BIND 8, typically twice as fast as BIND 9
- Sadly, does not work on FreeBSD 4.<8, OpenBSD (yet). Does work on Windows!

Lessons learned

- A database offers flexibility at the cost of memory and CPU requirements.
- The lack of zone (re)loading often offsets this
- Many people, author included, like zone files
- Benchmarketeers will not tune for performance!
  - they will also do their work on 48MB Pentium laptops using heavy handed databases
- Logging is way more expensive than doing DNS
- C++ was a great choice
  - none of the much feared performance & portability problems happened

Lessons learned 2

- It takes multiple years for a user base to grow
  - PowerDNS as a company is mostly defunct, but only now is the program taking off (1500 downloads/week)
- It takes even longer before useful external contributions start coming in (patches)
- Non-open source programs face a very tough sell
- Demand for DNSSEC is mostly an image thing ('yeah we do DNSSEC, we're secure')

THE BIGGEST LESSON LEARNED!

```
1<<31 – 1
!=
(1<<31) -1
```

General thoughts on DNS

- DNS is but the ARP of IP
  - 'layer 3 ARP'
- Except for IP addresses and MX, nothing important is in there
- DNS is **the** prime example of a robust distributed directory containing small data
- This is not due to the brokenness and limitations DNSSEC and other DNS extensions struggle with however
- The energy spent on DNS extensions could have generated DNS2 three times over!

Stuff we would keep & change

- Make an authoritative no such zone type
- Make an authoritative no such record type
- Make an authoritative no such type in this record type
- Add a signature field to all records
- Add ability to query multiple types at once
- Expand the ID space to 32 bits
- Replace label compression by generic compression

Stuff we would change & keep

- Allow clients to negotiate a secure context with a nameserver ('SSLDNS', hashcash)
  - So a stub resover can be secure too
- Add a zone (de)provisioning protocol
- Keep UDP
- Mandatory MTU path discovery
  - TCP is dog slow for small queries!
- Keep the binary format
- Keep serial numbers
- Add ability to delegate to IP addresses
- Add rsync-like zone modifications

Summarizing

- DNS could easily be usurped by IE, Exchange and Outlook doing a preferential search over at Microsoft for `enhanced information'.
- Protocol might stay the same but root-servers might be different!
- DNS is not well suited for enhancements (small packets, easily spoofed, very specific semantics)
- However, DNS remains the coolest protocol around! (with the possible exception of TCP)