Unbound

Validating Caching Resolver

Wouter Wijngaards
wouter@nlnetlabs.nl (NLnet Labs)

Overview

- Introduction: *Why another resolver?*
- Development History
- Features
- Implementation notes
- Testing
- Other DNSSEC tools from us
  - Nsd, Net::DNS, ldns, drill, autotrust

Introduction

- Why a new resolver?
  - Code diversity in DNS server monoculture
  - Alternative validator choice for BIND 9
- Deployment targets
  - Workgroup local DNS resolvers
  - Large caching resolver installations (ISP)
  - Validating library for applications
- About NLnet Labs
  - A not for profit, public benefit foundation
  - Developed NSD; DNSSEC aware, high performance authoritative name server

Development History

- The first architecture and a Java prototype was developed between 2006-2007.
  - Matt Larson, David Blacka — VeriSign
  - Bill Manning — EP.NET
  - Roy Arends — nominet
  - Jacob Schlyter — kirei
- NLnet Labs joined early 2007
  - porting the prototype to C and taking on maintenance.
  - First public development release on http://unbound.net/ in jan 2008

- Beta testing by:
  - Alexander Gall (switch.ch)
  - Ondřej Surý (.cz)
  - Kai Storbeck (xs4all.nl)
  - Randy Bush (psg, iij)
- 1.0.0 deployment sees uptake by BSD port trees and linux distros.

Features: Basic

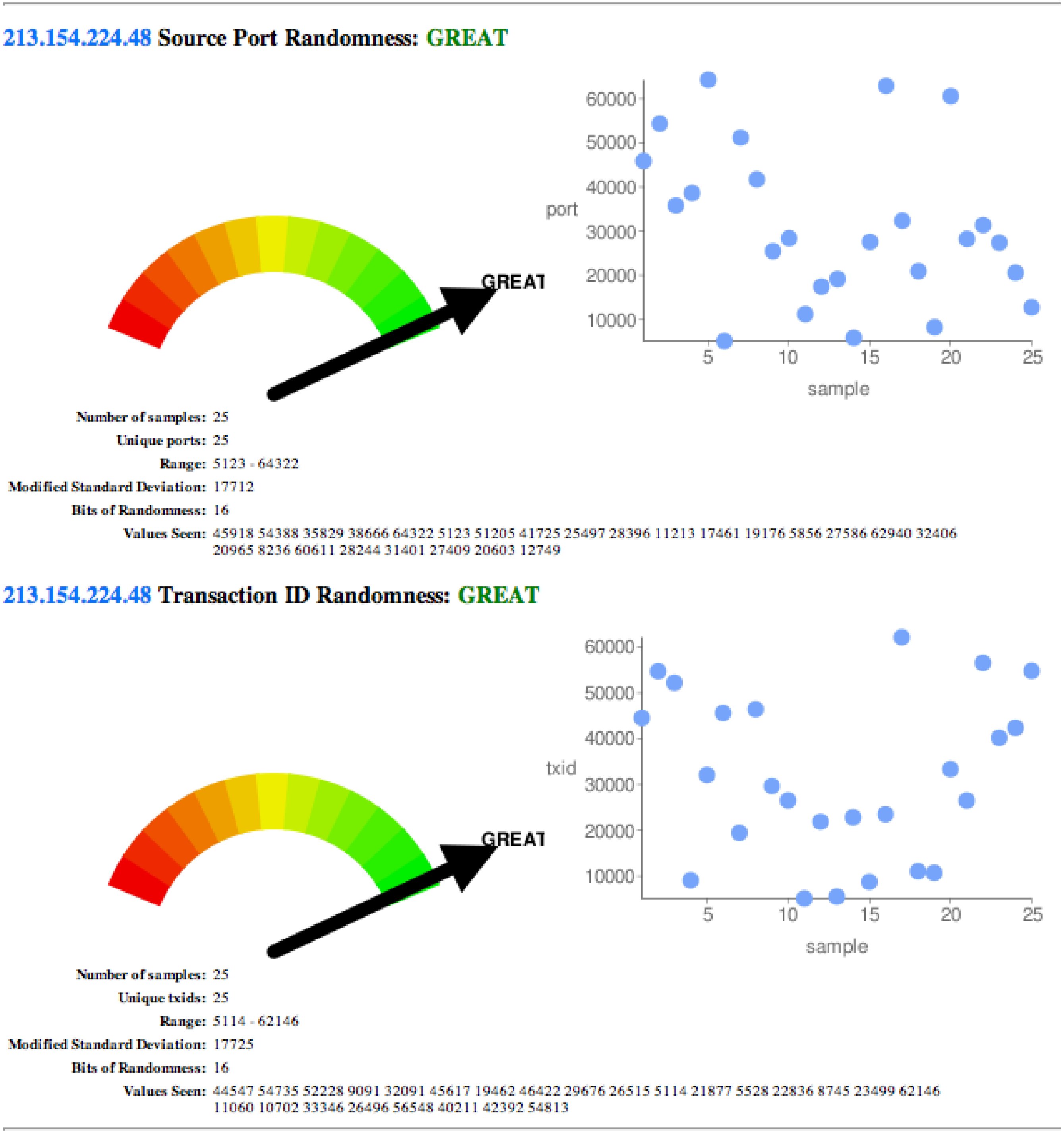
- DNS Server
  - Open source: BSD license
  - Recursion
    - IPv4 and IPv6 dual stack support
    - Access control for DNS service: not open recursor
  - DNSSEC validation
    - NSEC, NSEC3
- Tools
  - Unbound-checkconf
  - Unbound-host: validated host lookup
- Documentation
  - man pages, website and in code (doxygen)
- Thread support (optional): scalable performance

Features: Anchors and Authority

- Trust anchors: *feature rich*
  - – Rbtree for anchors – many islands
  - – DS and DNSKEY can be used for the anchor
  - – Zone-format and bind-config style key syntax
- Authority service: *minimal*
  - – Localhost and reverse (RFC1918) domains
  - – Can block domains
  - – Not authoritative server, use stub zones

Features: Paranoia

- Forgery resilience: *full featured*
    - Scrubber filters packets for out-of-zone content
    - Follows RFC2181 trust model
    - Follows all recommendations from dnsop draft
        - Query name matching
        - Strong random numbers for ID
        - UDP source port random
        - IP source address random
        - RTT banding



213.154.224.48 Source Port Randomness: GREAT

Number of samples: 25

Unique ports: 25

Range: 5123 - 64322

Modified Standard Deviation: 17712

Bits of Randomness: 16

Values Seen: 45918 54388 35829 38666 64322 5123 51205 41725 25497 28396 11213 17461 19176 5856 27586 62940 32406 20965 8236 60611 28244 31401 27409 20603 12749

213.154.224.48 Transaction ID Randomness: GREAT

Number of samples: 25

Unique txids: 25

Range: 5114 - 62146

Modified Standard Deviation: 17725

Bits of Randomness: 16

Values Seen: 44547 54735 52228 9091 32091 45617 19462 46422 29676 26515 5114 21877 5528 22836 8745 23499 62146 11060 10702 33346 26496 56548 40211 42392 54813

Implementation Notes On Forgery Resilience

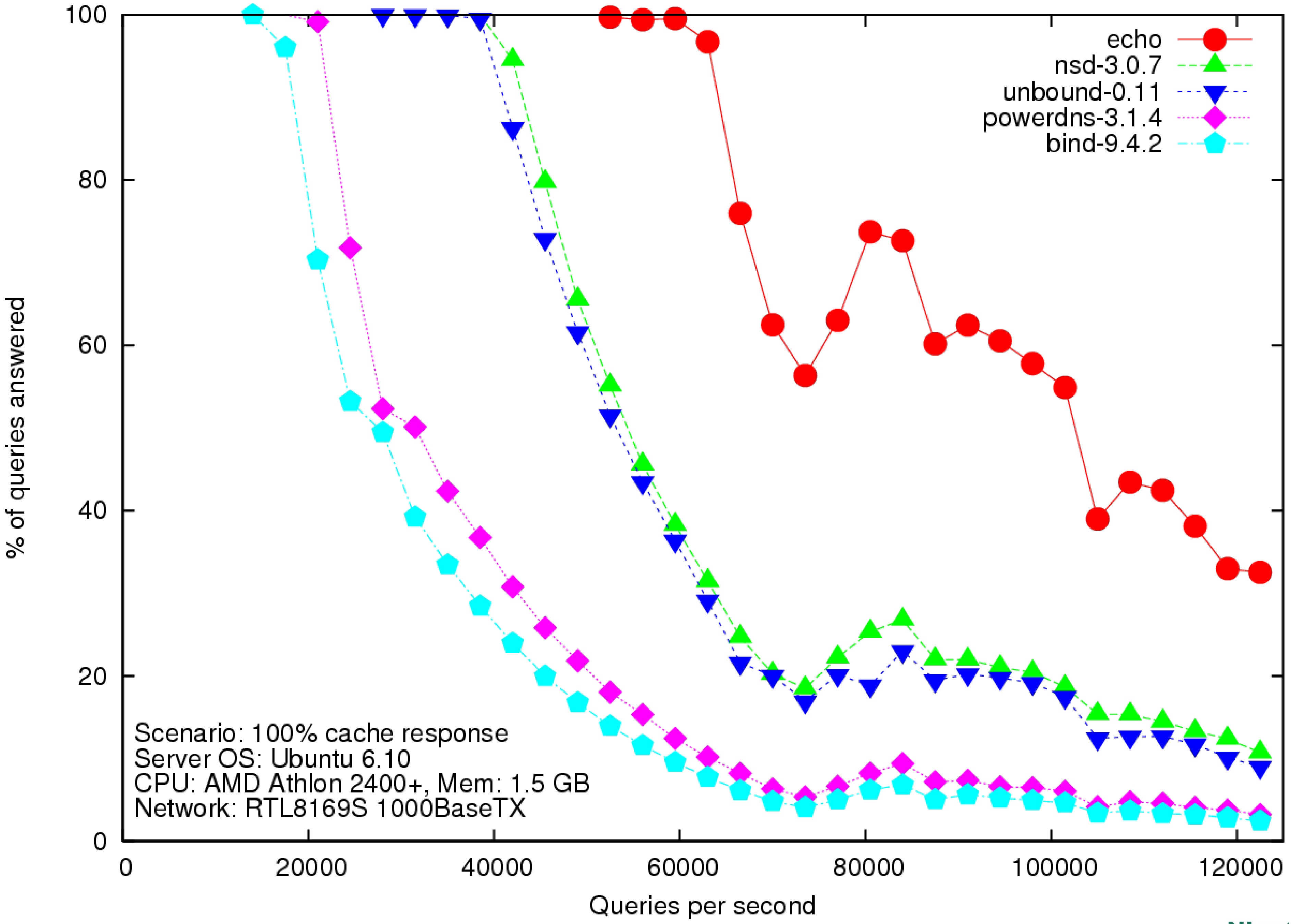
- Opening ports takes time
  - – The system calls, teardown and create new.
  - – The kernel becomes slower with more open fds
    - Opening 32000 ports at once slows down
- Hit limits of the OS and hardware
    - File descriptor limits
      - – Unbound estimates usage and fixes settings
    - Polling kernel functions
      - – Default to select()
        - » slow but portable
      - – Support libevent for nonportable epoll, kqueue
        - » Solaris (devpoll, evports) not well supported

Implementation Notes (ctd.)

- Problem
  - DNS resolver starts early in boot process
  - If grabs port other daemon wants, it fails
    - Error goes away after a reboot
- Avoid port conflicts with other daemons
  - Avoid <1024
  - Avoid IANA allocated ports (about 4-5k)
  - Avoid traditional unix ephemeral port ranges
    - So you can still ssh in case of trouble
  - User config of port ranges
  - Release a port immediately after use

Robust Code Development

- Regression tests
  - Unit testing of code
  - State machines tested on replay traces
  - Functionality tests (start daemon, make query)
- Beta tests
  - Test in the real world
- Performance tests
  - Cache performance
  - Recursion performance
    - Test against a known, stable environment


echo
nsd-3.0.7
unbound-0.11
powerdns-3.1.4
bind-9.4.2
% of queries answered
0
20
40
60
80
100
Queries per second
0
20000
40000
60000
80000
100000
120000
Scenario: 100% cache response
Server OS: Ubuntu 6.10
CPU: AMD Athlon 2400+, Mem: 1.5 GB
Network: RTL8169S 1000BaseTX

Other software

- NSD
  - Well known already
  - High performance authoritative server
  - IPv4, IPv6
  - DNSSEC, NSEC, NSEC3
  - TSIG, IXFR, AXFR

Net::DNS

- Net::DNS::SEC (with Ripe NCC)
  - Quick prototyping
  - Testing during IETF work
  - Tool box for DNSSEC roll-out
- Took over maintenance for whole of Net::DNS

LDNS

- Tool library
  - simplify DNS tools written in C
  - RFC compliant
  - IPv4 and IPv6 Support
  - TSIG Support
  - Online documentation, manuals
  - Inspired by Net::DNS

LDNS and DNSSEC

- Crypto based on OpenSSL
- NSEC & NSEC3
- SHA-256
- Hardware Signer Support
- Zone signer program

LDNS based: drill

- Like dig
- Debugging tool for DNS (SEC)
- Inspired the idea of LDNS
- Helped debugging NSD, BIND

More examples

- ldns-key2ds – Creates a DS record from a DNSKEY record
- ldns-keyfetcher – Fetches DNSSEC public keys for zones
- ldns-keygen – Generate private/pubkey key pair for DNSSEC
- ldns-read-zone – Reads a zone file and prints it with 1 RR per line.
- ldns-signzone – Signs a zone file according to DNSSECbis.
- ldns-update – UPDATE examples.
- ldns-walk – 'Walks' a DNSSEC zone
- ldns-zsplit – Splits a zone file in smaller parts
- ldns-zcat – Concatenates zone file parts split with ldns-zsplit
- ldns-compare-zones – See the differences between zones (added/removed names, added/removed rrs for names)
- ldns-revoke – set revoke bit on DNSKEY records
- And more

Autotrust

- RFC 5011 (draft-timers) implementation
- Under development
- Add-on to validator (Bind, Unbound)
  - Run from cron once per day, week
  - Writes trust anchor files
- Option to work with plain key rollover
  - No REVOKE bits published
  - Keep list of keys in missing state in check

Questions