OARC 26th Workshop
Madrid, Spain
14 May 2017

DNS-OARC Software

Jerry Lundström
Software Engineer

Software Development

- Use git, autoconf, automake, libtool, Semantic Versioning 2.0.0, conform to FHS 3.0, man-pages
- Continuous Integration using Jenkins, Travis-CI
- Coverity Scan for code analysis
- Compatibility testing on Debian, Ubuntu, CentOS, FreeBSD and OpenBSD
- Packages for Debian, Ubuntu (and CentOS)

DNS-OARC Software

- DSC – collect statistics from busy DNS servers
- DSC Presenter – explore the statistics
- dsc-datatool – convert, export, merge and transform DSC data
- dnscap – capture DNS traffic
- drool – replay DNS traffic
- dumdumd – drop traffic

DNS-OARC
Domain Name System Operations Analysis and Research Center

DNS-OARC Services

- Reply Size Test – resolver reply size / EDNS test
- Port Test – resolver randomize ports test
- DNS Entropy – resolver randomize the transaction ID test
- TLDmon – monitor TLD zones
- DANE Tester – DANE browser plug-in test
- Check My DNS – what the frick is your resolver doing!?

DNS-OARC
Domain Name System Operations Analysis and Research Center

DNS-OARC Libraries etc

- pcap-thread - PCAP helper library with POSIX threads support and transport layer callbacks
- omg-dns - library for parsing valid / invalid / broken / malformed DNS packets
- parseconf - configuration parser library
- sllq - Semi Lock-Less Queue
- Net::GetDNS – Perl bindings for getdns
- ripeatlas – Go bindings for RIPE Atlas API

DNS-OARC
Domain Name System Operations Analysis and Research Center

And now...

DSC

- Threads + fork() = BAD
  - System libraries detect usage of pthreads and start using mutexes but does not handle fork which in rare cases causes deadlocks
  - Threads default disabled as of v2.4.0

DSC

- Inconsistent statistics
  - Due to threads, weird select() behavior, interrupt when dumping reports and the wait before interval start

DSC

- Inconsistent statistics – SOLVED
  - Threads default disabled
  - “pcap_buffer_size <bytes>” to remedy kernel dropped packets
  - “pcap_thread_timeout <ms>” to control granularity of interval check (default 100ms)
  - “no_wait_interval” to skip the initial interval wait/sync

DSC

- To prevent “next time” - use Jenkins to continuously test DSC
  - DSC develop branch runs on all platforms and gets 10 QPS
  - 9 jobs monitors DSC, logs and XML output
- Let it run a week or so prior to release

https://github.com/DNS-OARC/dsctest

DSC

Special thanks to:

Anand Buddhdev, RIPE NCC

Klaus Darilion, NIC.AT

Vincent Charrade, Nameshield

DNS Replay Tool (drool)

- drool replays DNS traffic from packet capture files (PCAP) and sends it to a specified server
- Comcast sponsored project
- Released v1.0.0-beta.3 29th March
- Happily awaiting feedback, comments and/or thoughts...
  - How about a member-contributed sample PCAP repository?

DNS-OARC
Domain Name System Operations Analysis and Research Center

DNS Replay Tool (drool)

- Features include:
  - Utilize all the cores
  - Manipulate timing between packets to replay faster, slower or ignore (flood)
  - Loop packets infinitely or N iterations
  - Replay over UDP, TCP or as it was captured
  - … and more to come!

DNS Replay Tool (drool)

$ src/drool -vv -c 'text:timing ignore; client_pool target "127.0.0.1" "53"; client_pool skip_reply; client_pool sendas udp; context client_pools 3;' -r ~/dns.pcap

...

core info: runtime 0.160850035 seconds

core info: saw 286868 packets, **1783450**/pps

core info: sent 173686 packets, **1079801**/pps 39/abpp

...

Tested on Intel i7-6700K Ubuntu 14.04

DNS-OARC
Domain Name System Operations Analysis and Research Center

DNS Replay Tool (drool)

- Future improvements
  - Parse, match and add more statistics around the responses
  - Increase performance with configurable thread model and atomic queues
  - More statistics overall, control channel and GUI
  - Massive client IP simulation
    - Use the client IP from the capture, will require specific network setup

dumdumd

- High performance UDP/TCP server that ... just drops everything you send to it
  - Used during the development of drool to test the network code
  - Uses libev and/or libuv
  - Able to receive ~1 million UDP PPS using EV and ~1.1 million using UV (on an Intel i7-6700K Ubuntu 14.04)

RIPE Atlas API binding for Go

- Get Atlas measurements:
  - from JSON files
  - from Atlas RESTful API
  - from Atlas streaming API
- Measurement data structures
  - ping, traceroute, DNS, HTTP Get, NTP, SSLCert and Wifi

RIPE Atlas API binding for Go

```
18  func main() {
19      a := ripeatlas.Atlaser(ripeatlas.NewStream())
20      c, _ := a.MeasurementResults(ripeatlas.Params{"type": "dns"})
21      for r := range c {
22          print(r.DnsResult())
23          for _, s := range r.DnsResultsets() {
24              print(s.Result())
25          }
26      }
27  }
```

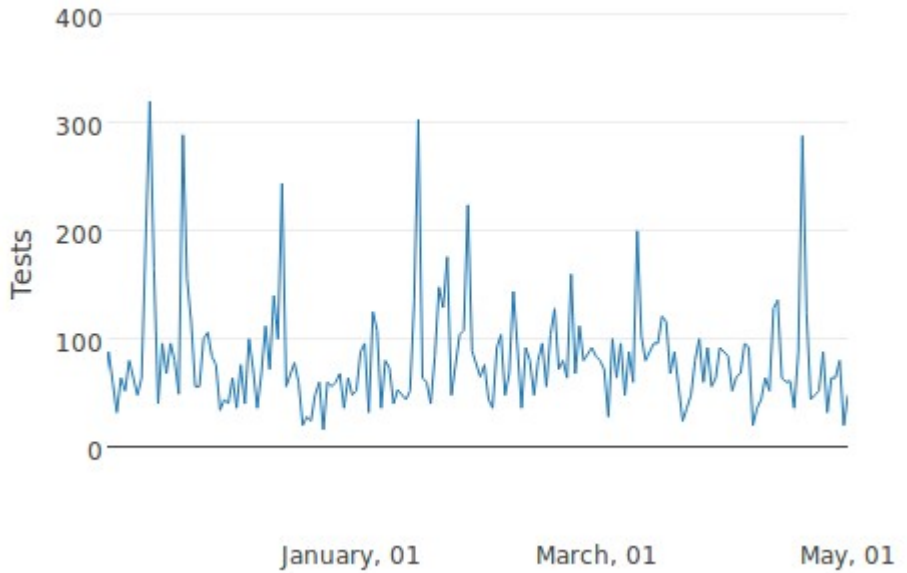
https://gist.github.com/jelu/ad8fd5d19bc43451e7f4fa3ae30ca9f4

DNS-OARC
Domain Name System Operations Analysis and Research Center

Check My DNS

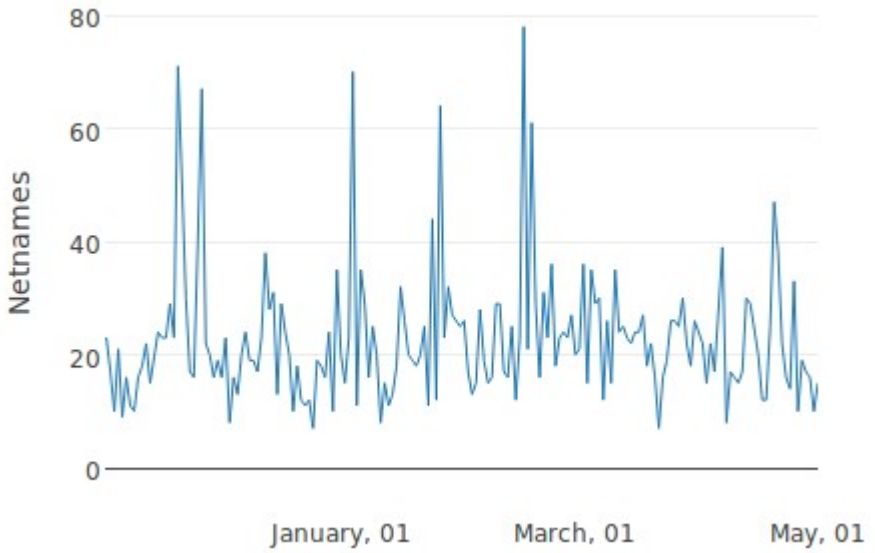
https://cmdns.dev.dns-oarc.net/

Check My DNS



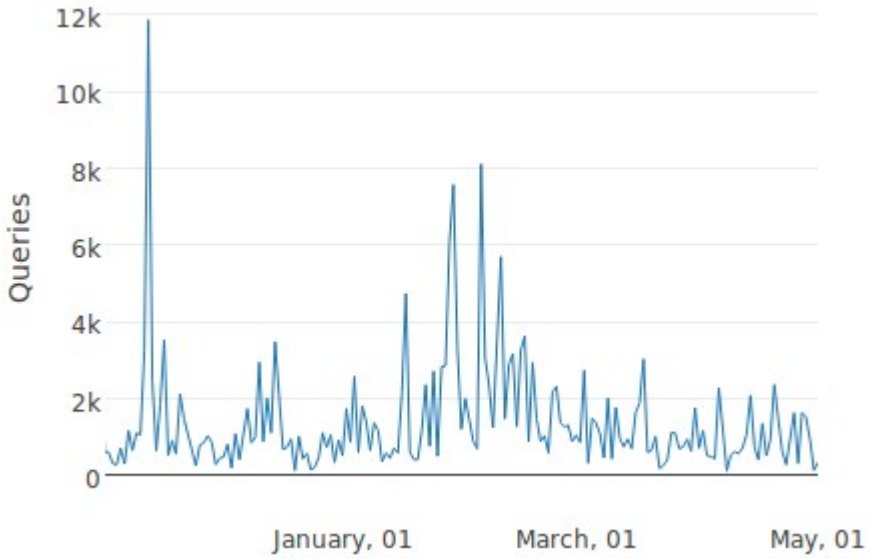
Number of tests / 4 tests per use = avg 20 uses per day

Check My DNS



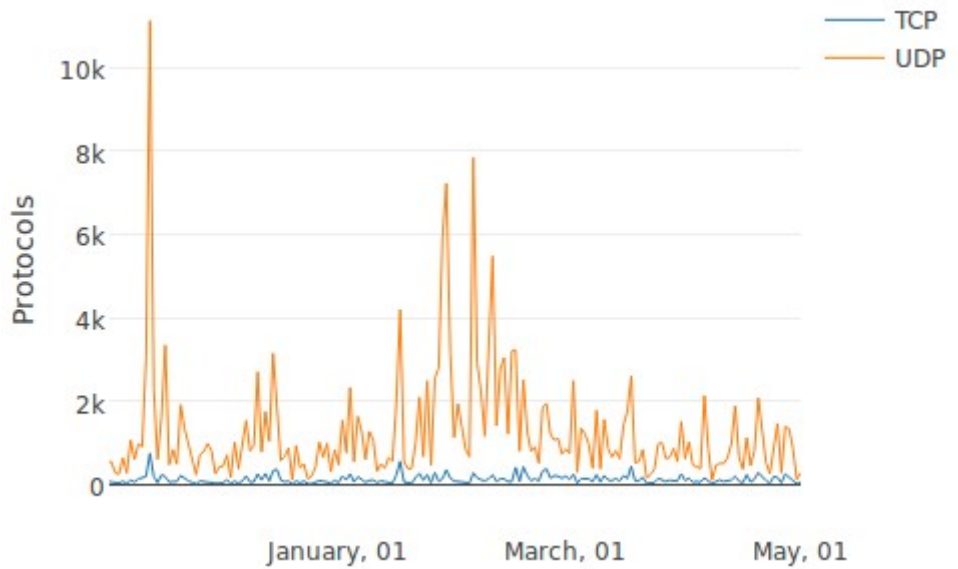
Number of unique netname found in WHOIS data

Check My DNS



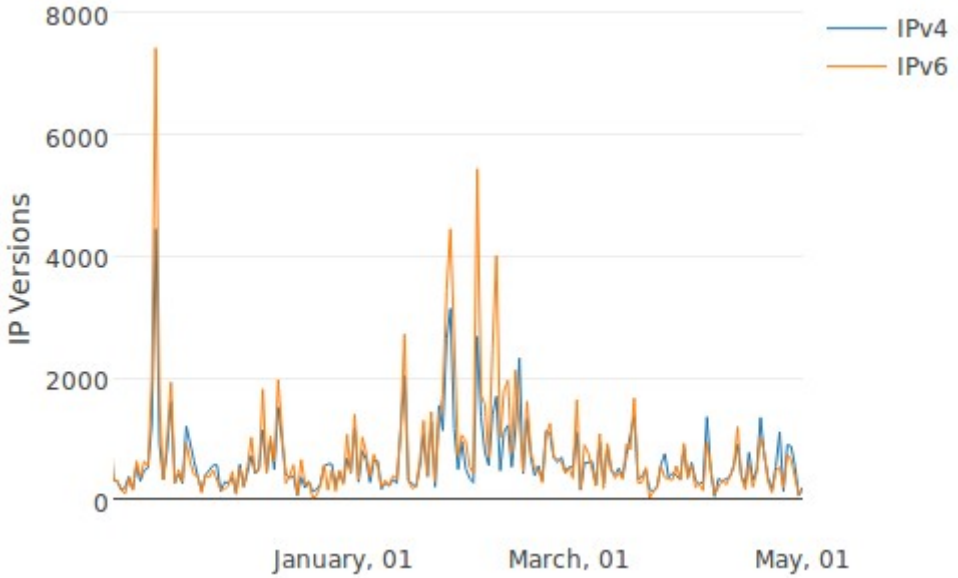
Number of queries seen at authority side

Check My DNS



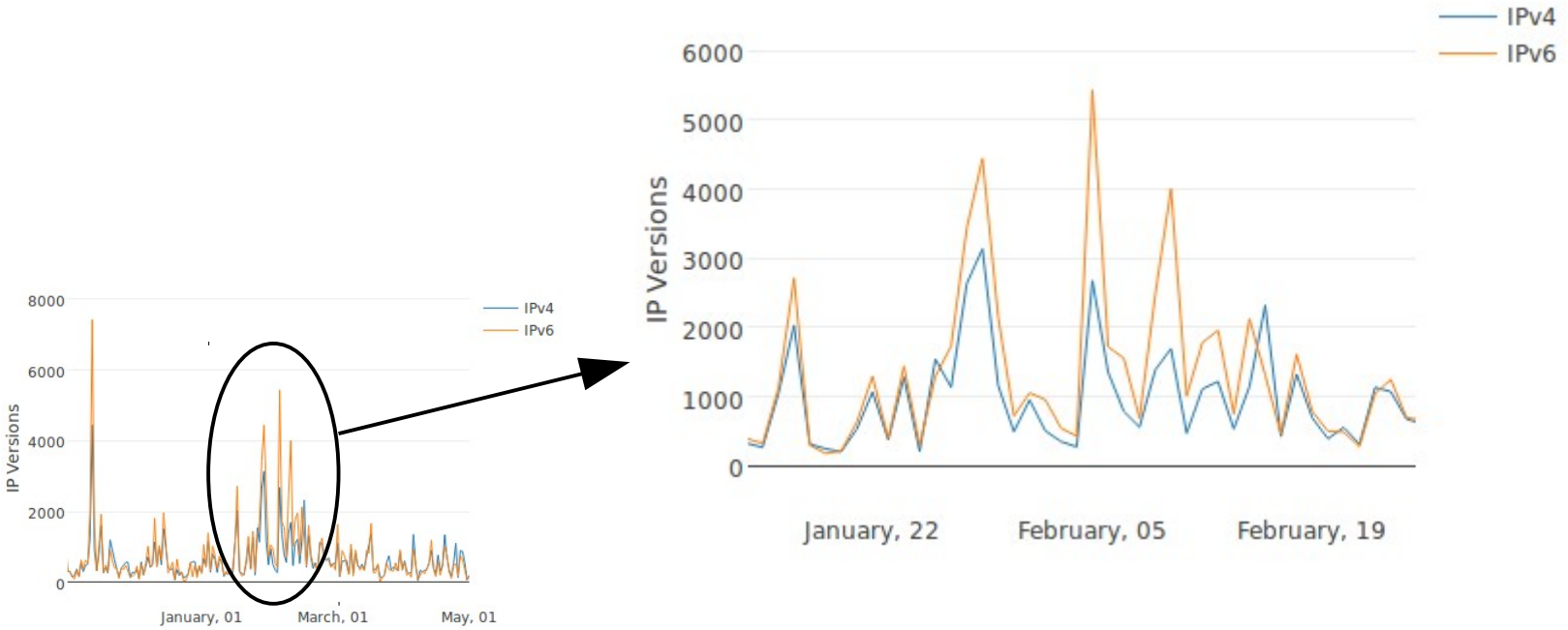
Number of queries per protocol

Check My DNS



Number of queries per IP version

Check My DNS



More IPv6 then IPv4! Yay!

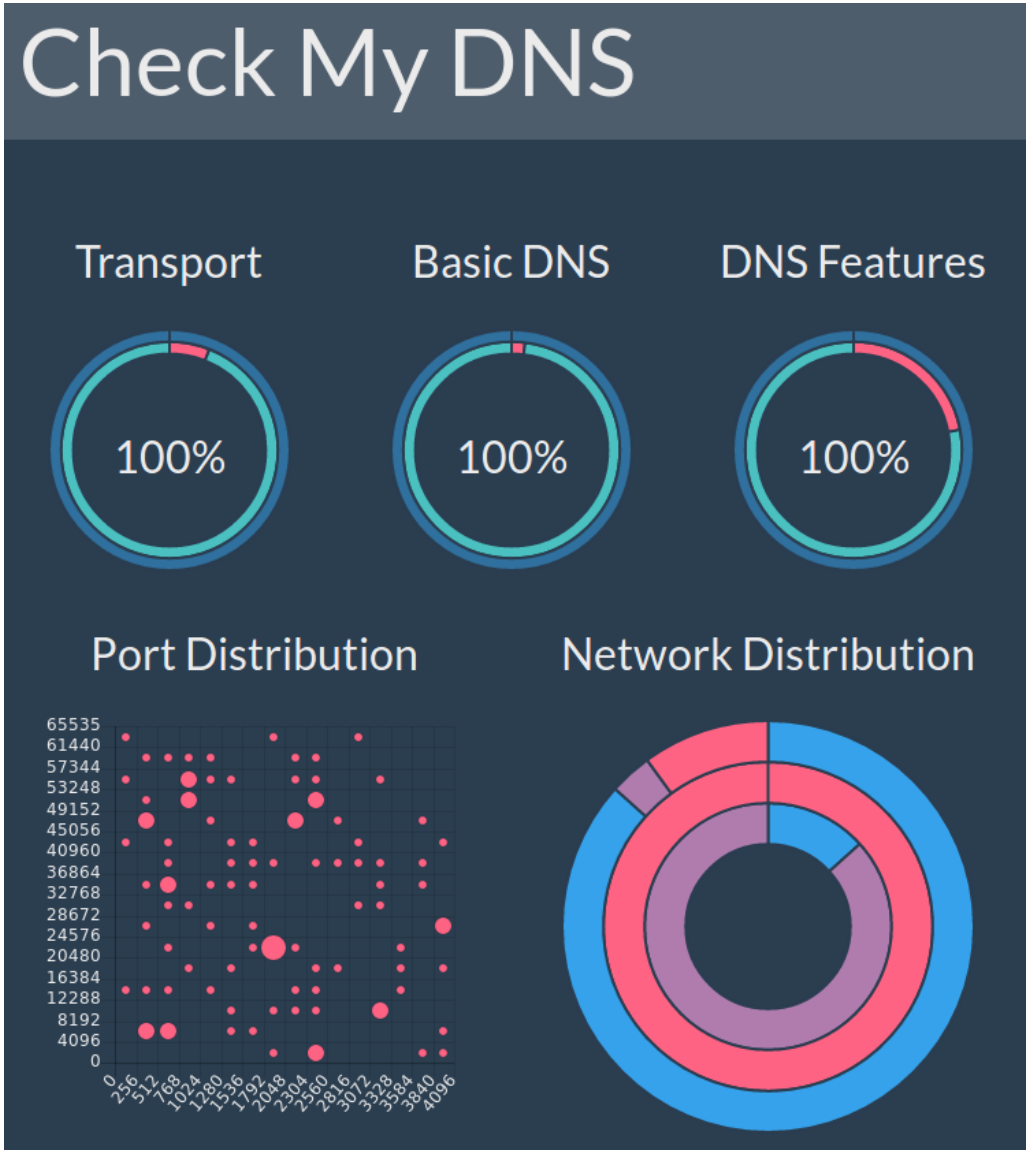
Check My DNS

https://cmdns.dev.dns-oarc.net/stats.html

- Remember:
  All test data, queries and responses are available for members to crunch!

Check My DNS

- Reimplementation in Go underway to increase performance from ~400 QPS to >50k QPS and rework API to make integration simpler
  - Make it easier to implement new tests
  - Have multiple point of presence
  - Run as a plug-in on any website to see how your visitors DNS resolvers operate



Disclaimer: colors are not final!

Q's?

https://github.com/DNS-OARC

https://www.dns-oarc.net/oarc/tools