OARC AGM 2021

Software Report

Jerry Lundström

Oct 15, 2021

Table of Contents

This report contains all major software happenings since OARC35 (May 2021). If you’re a frequent reader of my development update blog posts then you’ll probably recognize a lot of the information here.

1 Funded Projects

1.1 dnsperf - DNS-over-HTTPS

Last phase for this project was completed with the v2.7.0 release which added DNS-over-HTTPS support to both dnsperf and resperf.

This work was possible by funding from the Mozilla Open Source Support (MOSS) program and the Comcast Innovation Fund, many thanks for the support.

To recap, this project was divided into 3 phases and the first phase focused on removing the dependency of BIND’s internal development libraries. These libraries have been distributed along side BIND for many years as libraries for others to use, but recently that has changed and they are now only really for BIND (which is understandable because maintaining libraries like that costs a lot). Some of the functionality that dnsperf depended on has been added to dnsperf, for example parsing query names from datafiles. Other functionality have been covered by using LDNS, like construction of dynamic update queries. This first phase was concluded with the v2.4.0 release of dnsperf in December 2020.

The second phase refactored the network code into modules to make it easier to add the upcoming DNS-over-HTTPS support. It also added re-connection support for TCP and TLS along with new statistics around connections such as re-connections made and connection latency. This phase was concluded with the v2.5.0 release in March 2021.

Next and final phase added DNS-over-HTTPS support and this was concluded with the v2.7.0 release in August 2021. I hope this new functionality is useful for you and if you run into any problem then feel free to create an issue on dnsperf’s GitHub.

2 Highlights

2.1 Different RPKI OV solution

There have been some exciting development within the area of testing for RPKI origin validation. Willem Toorop and Jasper den Hertog (NLnet Labs) with help from RIPE NCC have created a new setup that works a bit differently than what Check My DNS used.

This new setup makes use of route announcement to announce a more specific route that is invalid. This makes it so that if you're not validating the Route Origin Authorization (ROA) object then you'll contact that IP address instead of going to the one with a valid ROA. As a references, the previous setup used by Check My DNS used two ROAs for separate networks, one valid ROA and one invalid ROA.

Here's an example where Willem tests from NLnog Ring:

```
nlnetlabs@xs4all01:~$ dig rpkitest.nlnetlabs.nl TXT +short
"HOORAY - Your resolver is protected by Route Origin Validation :)!"

nlnetlabs@isc01:~$ dig rpkitest.nlnetlabs.nl TXT +short
"NO - Your resolver is NOT protected by Route Origin Validation :("
```

Check My DNS was reconfigured to use this new setup in the beginning of July.

2.2 New dumdohd

While I was working on the reconnection code for DNS-over-HTTPS in dnsperf I wanted to test this using dumdumd. So I began adding DNS-over-HTTPS support into dumdumd using the same library, nghttp2, but I quite quickly saw that this would take a lot longer then I had time for.

This made me look at alternatives and once I found the client/server example in nghttp2's repository I took the server part and started to modify it to work as dumdumd. That gave birth to dumdohd which is a tool that takes incoming HTTP2 requests and reflects them as responses, copying the data from the request into the response. It can also be configured to randomly drop the connection on a request to mimic bad networks in the same way dumdumd can.

You can find dumdohd in dumdumd's repository.

2.3 DSC+Grafana crash course

Been doing a few crash course on how to setup DSC, dsc-datatool, InfluxDB and Grafana, and create some basic graphs. These crash courses aren't what you'd think a traditional course or training session would be. Instead I want them to be much more interactive, kinda like show & tell / discussion etc.

Basically it's me sharing my desktop and showing you how to install and setup all the software, this is done on a test instance of Check My DNS (debian 11). Then I will show how to create some Grafana graphs using data from DSC. After that everyone gets access to the Grafana instance where they can start creating their own graphs while asking questions.

These sessions are split up into 3 parts with short breaks in-between:
- part 1 ~20min) install and setup software, create simple QTYPE graph
- part 2 ~15min) reconfigure dsc to measure DNS response time, create graph for response time, create RCODE alerts graph for NOCs
- part 3) create your own graphs, ask questions and discuss

As of writing this I've ran 3 sessions for individuals, up to 6 persons per session, and have scheduled 5 team sessions in the coming months.

If this peaked your interest and you haven't already signed up then let me know!

3 Software Updates

A key part of DNS-OARC's mission is to develop, maintain and host various software tools for DNS data collection, measurement and analysis. OARC can develop new, or enhance features of existing, tools via a custom for-hire development contract. OARC Members will receive priority for such work, and at a discounted rate depending on their membership tier.

You can find a list of all our software and information about funded development here:

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

3.1 dnsperf

dnsperf and resperf (part of dnsperf) are tools that makes it simple to gather accurate latency and throughput metrics for DNS services. These tools are easy-to-use and can simulate typical Internet usage, so network operators can benchmark their naming and addressing infrastructure and plan for upgrades.

While most of these releases are related to the DNS-over-HTTPS project, other changes have also been done and they are listed here.

v2.6.0

Added EDNS options parameter -E to resperf and a script for generating EDNS Client Subnet options (see contrib/ecs-gen).

v2.7.0

Added DNS-over-HTTPS support!

DNS-over-HTTPS can be used by specifying transport mode doh and you should also look at the dnsperf(1) man-page (or -H) for the extended options doh-uri and doh-method, which controls aspects of DoH/HTTP/2 that you might want to set.

Other fixes:

- Added check when constructing DNS packet so that total length of labels does not exceed 255 bytes
- Fixed connection/reconnection state handling for DoT transport
- Fixed event handling by initializing them directly when opening the sockets, otherwise events could have been missed which would give incorrect statistics

v2.7.1

Fixed issues with constructing wire-format DNS when the domain names includes escaped characters such as \123 or \..

3.2 dnsjit

dnsjit is a combination of parts taken from dsc, dnscap, drool, and put together around Lua to create a script-based engine for easy capturing, parsing and statistics gathering of DNS messages while also providing facilities for replaying DNS traffic.

v1.2.0

Added development files that can be installed and used to create custom modules for dnsjit or stand-alone tools using dnsjit.

In examples/ you'll now find:

- modules/input-example: This input example is based on dnsjit.input.zero which was a testing modules during the early days of dnsjit. It's a C module that generates empty objects for the receiver.
- modules/filter-example: This filter example is C module that counts the number of objects passed to it before sending it to the receiver.
- modules/output-example: This output example is based on dnsjit.output.null which was a testing modules during the early days of dnsjit. It's a C module that will just discard objects it receives.
- modules/lib-example: This example Lua module takes two core_timespec_t C objects and gives the duration between them as a string.
- stand-alone-tool: This example is based on test_pcap_read.lua and test_throughput.lua which was previous located in examples/.
  There are two installable Lua programs and shows how to depend on a dnsjit version, depend on specific dnsjit modules and how to run tests using make test.

All these examples can easily be copied and renamed to build and distribute your own dnsjit modules and tools, using autotool for configure, make and make install.

Development files will also be installed, or can be installed via dnsjit-dev/dnsjit-devel packages. All C headers have been prefixed with dnsjit/ (for example #include <dnsjit/version.h>).

Thanks to this new setup, the module output.dnssim has been moved out from dnsjit and placed in DNS shotgun's repository.

New modules:

- Added input.zpcap, module for reading LZ4/ZSTD compressed PCAPs
- Added core.loader, module for loading C modules using LuaJIT's ffi interface and package.cpath
- Added core.file with core_file_exists(), a C function to check if a file exists

Other changes:

- Add <dnsjit/version.h> for DNSJIT_MAJOR_VERSION, DNSJIT_MINOR_VERSION, DNSJIT_PATCH_VERSION
- dnsjit: Remove version print on start
- dnsjit.input.zero: Will require("example.input.zero") for backwards compatibility
- dnsjit.output.null: Will require("example.output.null") for backwards compatibility
- core/timespec: Add :max_init(), return a new object with maximum values set for seconds and nanoseconds.
- output/pcap:
  - Update open() man-page, indicate usage of pcap_dump_open()
  - Add have_errors() to check for write errors during/after dumping
- input.fpcap: Add fadvise_sequential() to advise sequential read of the file
- examples/dumpdns.lua: Add support for dumping compressed PCAPs

Bugfixes:

- lib/getopt: Fix short options, error if length is not 1
- core/timespec: Fix typo in struct documentation

v1.2.1

This patch release fixes packages for SLE/openSUSE and the generation of an environment script for the stand-alone tool example.

3.3 drool

drool can replay DNS traffic from packet capture (PCAP) files and send it to a specified server, with options such as to manipulate the timing between packets, as well as loop packets infinitely or for a set number of iterations. This tool's goal is to be able to produce a high amount of UDP packets per second and TCP sessions per second on common hardware.

v2.0.0

This major release is a complete rewrite of drool to Lua using dnsjit!

This includes two commands:

- “drool replay”: replay DNS traffic from packet capture (PCAP) files and send it to a specified server, with options such as to manipulate the timing between packets, as well as loop packets infinitely or for a set number of iterations.
- “drool respdiff”: replay DNS queries found in the PCAP, but only if a correlating response is also found. The query, original response and the received response is then stored into a LMDB database which can then be used by respdiff by CZ.NIC for analysis.

See “man drool”, “man drool reply” and “man drool respdiff” for more information.

3.4 dnsmeter

dnsmeter is a tool for testing performance of a nameserver and the infrastructure around it. It generates DNS queries and sends them via UDP to a target nameserver and counts the answers. Sender addresses can be spoofed from a given network or from the addresses found in the PCAP file.

v1.0.2

This release fixes an issue with source port being static when only using -q to generate traffic from one host/IP. The source port is now randomized for every DNS query.

Other changes is mainly about build system, packages and fixed issues detected by code analysis tools.

4 OARC Portal Updates

v1.2.5

Fixed an issue with the confirm email feature which was related to the a datetime comparison that was incorrect and resulted in a 500 Internal Server error.

v1.3.0

Added voting representative feature, you can now select one contact that will vote during OARC AGM, and updated the welcome letter which gets sent to new contacts.

Fixed display issue for contact state, it will now disable active/hidden for disabled/deleted contacts if the contact limit has been reached.

v1.3.1

Fixed cut&paste mistakes from v1.3.0.