OARC35

Software Report

Jerry Lundström

Apr 16, 2021

Table of Contents

This report contains all major software happenings since OARC's 2020 AGM back in September. 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 RPKI Origin Validation Visibility

With the funding awarded by the ARIN Community Grant Program in October 2020, Check My DNS has been given some much needed updates and more!

Core Updates

Major updates have been done to the core of Check My DNS. This included the Go version, all Go dependencies, jQuery, Bootstrap, ChartJS and the theme from Bootswatch.

RPKI Origin Validation Checks

While these checks were added back in November 2019, they were not enabled by default because there was no way to display them properly.

With this work, these checks will use the new Achievements (see below) and in so are enabled by default. But there are important caveats that Job Snijders mentioned when we added this:

> *While this test will work reliably in most use cases, there are a few caveats to consider when interpreting the results of this test.*
>
> 1. *Any resolver that is hosted in a network that points a default route to NTT / AS 2914, or is downstream of a network pointing default to NTT, will be able to reach the RPKI invalid beacons at NTT.*
>   *This means that even if the resolver operator applies RPKI based BGP Origin Validation (RPKI OV) on all their peering routers and rejects invalid route announcements (which is good!) the test will indicate that no RPKI OV is happening: _a false negative_.*
>
> 2. *If the resolver is hosted in a network that is using another intermediate network to reach NTT, and the "in between" network is doing RPKI OV, the test may result in a false sense of security as the resolver network isn't doing RPKI OV but the intermediate network is! Depending on the circumstances this is less optimal or fine.*

See full blog post "RPKI origin validation for resolvers!" for more information.

Achievements

Added achievement badge/shield icons for RPKI origin validations over IPv4 and IPv6.

These achievements can be used to indicate features and functionality, or a collection of them, that might be outside the scope of the rating. For example, the RPKI origin validation checks do not currently affect the rating you get, even if they fail.

This is also a very easy visual way (thanks freesvg.org!) of showing support for a specific feature or functionality. Another example – that I would like to add – is checking for all DNSSEC algorithms' support. Here, the check could be that you support enough DNSSEC algorithms to have a functional DNS but if you support all the current algorithms you could also be awarded an achievement for it!

Still Beta?

The beta banner has been removed because it was quite a difficult UI element to maintain. This does not mean Check My DNS is now a super duper stable production-ready thing, it is still mainly a research and testing tool.

1.2 dnsperf - DNS-over-HTTPS

This ongoing project, which is funded by Mozilla Open Source Support (MOSS) program, and the Comcast Innovation Fund, began in October 2020 and is split up into 3 phases.

The project aims to add DNS-over-HTTPS support to dnsperf, but doing so required additional work to be done beforehand.

Please note: all references here to dnsperf also includes resperf, as they come from the same source repository.

Phase 1 – BIND development libraries (Completed)

The BIND development libraries, those libraries used internally in BIND, has long been required to build dnsperf.

Back when dnsperf was created, these libraries were distributed so that other software could use them. But recent change to how BIND is maintained, most likely due to the complexity these libraries created, have marked these libraries as internal-only and not meant for other software to use.

This became noticeable in the spring of 2020 when dnsperf builds started failing because of the roll out of BIND 9.16. Thanks to a patch by Petr Menšík (Red Hat), we got a work-around for dnsperf v2.3.4 but that was not going to last long term.

So the task for the first phase was to remove this dependency, which was done successfully and released in v2.4.0.

Phase 2 – stateful connections (Completed)

Up until very recently dnsperf only supported DNS over UDP but with the release of v2.3.0 in July 2019, both TCP and DNS-over-TLS support was added.

While this implementation was successful, it was added upon a connection-less workflow which made stateful connection handling quite difficult to work with. When we were planning the work for this project it became clear that this had to change in order to add support for more protocols.

The idea was to have one thread processing the datafile, generate queries and processing responses. Another thread was going to use libuv for the communication and to send information between the threads Concurrent Kit’s rings were picked.

While doing this work I ran into a whole lot of strange problems, that I saw could easily eat up all the allotted time for this phase without actually getting anywhere. So I had to make a decision, and I decided to go back to the old code and try to re-factor it so that multiple protocols were easier to manage and so that re-connection support could be added.

This chosen approach was successful, and with version v2.5.0 both dnsperf and resperf now have a more modular engine and re-connection support for TCP and DoT!

Phase 3 – DNS-over-HTTPS (Upcoming)

The last phase of this project is to add support for DNS-over-HTTPS. This is scheduled to begin in May and be finished by end of August.

2 Highlights

2.1 Mattermost OARC Software channel

On our Mattermost chat platform for the DNS Operations Community and OARC Members, there is now an "OARC Software" channel to talk about OARC's software, its features, and for us to announce new releases.

It is also a place where you can interact with others using our software and get help (as a compliment to some of our mailing lists) and is open for anyone to join.

2.2 New major dnscap release

The v2.0.0 release contains three backward incompatible changes, two new command line options and a completely restructured man-page, please read the change notes carefully before upgrading!

The first backward incompatible change has to do with the removal of libbind dependency. This library was causing segfaults on OpenBSD due to shared (and overwritten) symbols with OpenBSD's libc. It was replaced with LDNS and LDNS renders domain names as Fully Qualified Domain Names (FQDN, the trailing dot!) so every output of a domain name has been changed to a FQDN. This also changes `-X`/`-x`, which will now match against FQDNs.

The second backward incompatible change is that `-6` has been removed. This was used to alter the BPF in order make it work for IPv6, dnscap adds specific filters to IP and UDP headers which does not work for IPv6 traffic. The generated BPF has been changed to allow IPv6 to always pass, making the option obsolete. IPv6 filtering is then done in dnscap.

The last backward incompatible change has to do with the output format of `-g` related to EDNS0 and is now more consistent with the rest of the parsable output:

- No more spaces in the output
- Fix incorrect \ and extra empty new-line
- All EDNS0 options are added after `edns0[...]` using comma separation, example: `edns0[],edns0opt[],...`
- Client Subnet format: `edns0opt[ECS,family=nn,source=nn,scope=nn,addr=...]`
- Unknown/unsupported code: `edns0opt[code=nn,codelen=nn]`
- Parsing error messages have changed, they came from libbind, now comes from LDNS

2.3 New features in dumdumd

dumdumd is a small piece of software that was created to be able to test other software’s network code. It was first designed to just drop everything it received but has lately also been given reflection capabilities and support for TLS.

When working on the re-connection code for dnsperf, I added two types of forced disconnect for TCP and TLS. The `-C` option will make it so it closes the connection after receiving the first message, and after reflecting it if that is enabled. The `-D nnn` will do random disconnect when receiving, based on the given percentage. So for example, `-D 50` will drop about half of all connections.

This has helped immensely when making sure that the re-connecting code in dnsperf works as it should during a less then optimal network situation, and I hope that it can be helpful for others too.

2.4 DSC Grafana example dashboards update

The example graphs that show PPS/QPS has been updated and the numbers should be correct now. They have been changed to display `field(value) sum() math( / ($_interval_ms / 1000) )` which now also should work correctly when selecting a different Time Resolution of the graph.

> ***Please Note!***
>
> *The setup and graphs described in the Wiki are meant as an example of what can be done. It’s important to understand and learn how DSC’s datasets work, how they are exported to InfluxDB and how graphs in Grafana works.*
>
> *Grafana is an amazing tool, and you will greatly benefit in customizing your own graphs for your own purpose and needs.*

For details on how to setup your own test instance, please see this Wiki:
https://github.com/DNS-OARC/dsc-datatool/wiki/Setting-up-a-test-Grafana

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 dsc

DNS Statistics Collector (dsc) is a tool used for collecting and exploring statistics from busy DNS servers. It uses a distributed architecture with collectors running on or near name-servers sending their data to one or more central systems for display and archiving.

v2.11.2

Fixed a bug in `asn_indexer`, it was not enabled correctly when using MaxMindDB for lookup so all lookups would be set to unknown.

3.2 dsc-datatool

dsc-datatool is a tool for converting, exporting, merging and transforming dsc data using a plugin architecture. It can be used to convert dsc XML data into InfluxDB which can be used by Grafana to display DNS statistics.

v1.0.2

Fixed a bug in DAT file parsing that could lead to a crash.

3.3 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.

v2.4.0

The transport mode option `-m`/`-M` now recognizes `dot` alongside `tls` for DNS-over-TLS.

Added `-W` for outputting warnings and errors to stdout.

Fixed a potential memory leak of query descriptions when using verbose and only use TLS v1.2 and above for DoT/TLS.

v2.4.1

Fixed an issue with the socket readiness function that could cause a buffer overflow (for example `-T 10 -c 2000`) due to `select()` being limited to check 1023 sockets. Changed to use `poll()` which has no limit.

Fixes to the contrib script `queryparse` that has to do with python v2 and v3 compatibility and better exception handling.

v2.4.2

Fixed a few issues with reading of the datafile which could lead to "ran out of data" errors.

The problem was that reading from the datafile was done before finding a socket to send it on, or before checking socket readiness, and that lead to progressing the queries without really doing anything. Another issue was that if the reading of lines from the datafile perfectly aligned with the buffer, it would be treated like EOF and caused an exit.

v2.5.0

Improved the tracking of socket readiness while connections are establishing, previously this could lead to a lot of warnings about "socket not ready" which should now hopefully be less frequent.

`resperf` now tries to process more requests each run to hopefully not hit max outstanding so easily when using high QPS settings.

Two new `resperf` options:

- `-R`: Reopen the datafile if it runs out of data before the testing is completed. This allows for long running tests on very small and simple datafile.
- `-F <fall_behind>`: Sets the maximum number of queries that can fall behind being sent. `resperf` will stop when this limit is reached and it can be relatively easy to do so if `-m <max_qps>` is set too high. The default is 1000 and setting it to zero (0) disables the check.

Bugfixes:

- Fixed port handling for host/network format when setting client side port with `-x`.
- Fixed support for quoted characters, `\000` and `\.`, in domain names, this was lost when removing BIND's internal development libraries.
- Fixed issue in `dnsperf`, it would loop forever if no connection could be established.
- Fixed potential buffer overrun in `resperf` when using response id for `queries[]`.
- DoT: Fixed bug when sending from buffer.

v2.5.1

Re-added support for `TYPEnnn` and `ANY` query types in the datafile, this was missed during the removal of the dependency on BINDs development libraries in v2.4.0.

v2.5.2

Tweaks to the reconnect code for TCP and DoT.

For TCP, atomic operations are used to signal the need to reconnect from the receiving thread to the sending, as the sending is the one in charge of reconnecting. This speeds up detection of connection lost which reduces the amount of lost queries on a disconnect.

This change does not affect DoT as much, as the SSL context shared between the threads are protected by a mutex. But a bug was found in `sendto()` for DoT that could drop a query if the socket was busy sending.

The connect and reconnect socket events have been split into connecting, connected and reconnecting, reconnected. This is to report more correct reconnect events when it comes to DoT, because the connection could be lost while negotiating TLS.

3.4 dnscap

dnscap is a network capture utility designed specifically for DNS traffic. It produces binary data in pcap(3) and other formats. This utility is similar to tcpdump(1), but has a number of features tailored to DNS transactions and protocol options. DNS-OARC uses dnscap for DITL data collections.

v1.12.0

Fixed handling of `-?` option for dnscap and all plugins.

The rzkeychange plugin got a new option `-D` for dry run mode (mainly for testing) and fixed handling of `-a`. It will now also give an error if too many `-a` are used.

v2.0.0

Beside the notes in Highlights about this major release, additional new features and fixes were made.

New options `-q` and `-Q` to filter on matched/not matched QTYPE.

Fixed a memory leak in EDNS0 ECS address parsing and a potential `memcpy()` of a null pointer.

Fixed CBOR output inclusion build checks since LDNS is now always available. Added macros for Apple and Windows endian functions. Restructured and corrected the man-page.

v2.0.1

Fixed an incorrect line break in the eventlog plugin that would otherwise make the output invalid.

3.5 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.

v1.1.0

Added a new module for handling Base64 URLs, new call for error handling, new call for opening PCAPs using file descriptors and bug fix for `lib.getopt`.

The `dnssim` module, heart of DNS shotgun, now has its own version and changelog, this is to prepare it for being moved outside of dnsjit's repository in the future.

New modules, calls, features:

- New `lib.base64url`: Utility library to convert data to base64url format
- `core.log`: New call `Log.errstr()`: Convert error number to its text representation
- `input.fpcap`: New call `Fpcap.openfp()`: Open a PCAP file for processing using a file descriptor, for example `io.stdin`
- `output.dnssim`: Support for DNS-over-HTTPS

Bug fixes:

- `lib.getopt`: Fix bug where `-` and `--` could not be used as arguments to options

Other changes:

- Fix typo in configure help text
- Add coverage
- `filter.ipsplit`: Extend PRNG modulus to 2^31, new implementation is the same as glibc's `rand()`
- `lib.ip`: Fix typo in documentation
- `output.dnssim`:
    - This module now has its own changelog
    - Updated to v20210129
    - Depend on libhttp2 for dnssim DNS-over-HTTPS capabilities
- `output.pcap`: Log libpcap error when failing to open
- SUSE packages now depend on moonjit because of lack of LuaJIT support

3.6 PacketQ

packetq is a command line tool to run SQL queries directly on PCAP files, the results can be outputted as JSON (default), formatted/compact CSV and XML.

v1.4.3

Added support for the DNS resource record types SVCB and HTTPS.

3.7 tinyframe

tinyframe is a minimalistic library for reading and writing the Frame Streams protocol which is used to encapsulate DNSTAP.

This library is currently a Work in Progress which means backwards incompatible changes can be made. Once version 1.0.0 is released it will follow Semantic versioning 2.0 as all other software does.

v0.1.1

Update to `tinyframe_write_control()`, it now checks for valid content field types.

3.8 dnswire

A C library for encoding/decoding different DNS encapsulations and transporting them over different protocols. It currently supports DNSTAP using Protobuf sent over Frame Streams using tinyframe.

This library is currently a Work in Progress which means backwards incompatible changes can be made. Once version 1.0.0 is released it will follow Semantic versioning 2.0 as all other software does.

v0.2.0

Fixed `dnstap_decode_protobuf()`, was setting socket family and protocol incorrectly if the types were not supported by dnswire.

The `enum dnstap_message_type` had a typo in the unknown type name, it’s was familt instead of family.

Fixed a bug in `dnswire_writer_set_bufsize()`, changing buffer size while having something in the buffer was done incorrectly.

4 Member software updates

4.1 DNS Shotgun

DNS Shotgun is a benchmarking tool specifically developed for realistic performance testing of DNS resolvers. Its goal is to simulate real clients and their behavior, including timing of queries and realistic connection management, which is important for measuring DoT and DoH.

The second version v20210203 brings a new user interface, mixed-protocol scenarios and IPv4 support. With the new user interface, it is possible to simultaneously simulate the traffic over multiple different protocols – e.g.:

- 80% of clients use UDP
- 10% of clients use DoT
- 10% of clients use DoH

The project was updated with a comprehensive documentation describing the used methodology and recommended usage. You can also read more in this recent blog post.

– Tomas Krizek (CZ.NIC)

4.2 flamethrower

Flamethrower is a small, fast, configurable tool for functional testing, benchmarking, and stress testing DNS servers and networks. It supports IPv4, IPv6, UDP, TCP, DoT, and DoH and has a modular system for generating queries used in the tests.

- An experimental DNS over QUIC module is under development. The goal is to aid in testing for server developers wishing to implement the draft RFC. We are seeking those interested in helping with development and/or testing.

– Shannon Weyrick (NS1)

4.3 pktvisor

pktvisor is an open source observability tool that collects and summarizes real time network and DNS statistics and provides both a command line UI as well as centralized collection to your TSDB of choice + Grafana. Summarized information includes, for example: packet rate percentiles, counts by protocol and IP version, DNS rcodes, qtypes, transactions, top 10 heavy hitters of IPs, ASNs, Geo, DNS qnames, slow transactions, and more

- Version 3.1.0 saw a major refactor to modularize the input and analyzer subsystems, paving the way for upcoming input methods like dnstap and sflow/netflow (in addition to the current packet capture). Modules may be dynamically loaded at runtime. A full admin REST API was introduced to control capture and analysis dynamically.
- Version 3.2.0 introduces support for native Prometheus metric output, and easy install and collection via docker container or static linux binary.

– Shannon Weyrick (NS1)