DNS Monitoring

with Prometheus and dns_exporter

DKNOG 14, March 7th, 2024

Thomas „Tykling” Rasmussen

On the menu

- Introduction
- Idea
- Installation
- Configuration
- Metrics
- Dashboards
- Bugs
- Questions

Introduction

- Tyk-of-all-trades, mostly Python programmer @ dayjob
- DNS nerd – I run UncensoredDNS (since 2009)
- Prometheus is my monitoring system of choice (I talked about it 5 years ago at DKNOG9!)
- I have a healthy interest in infosec, privacy and encryption
- I want better monitoring of DNS (servers and names)

Idea

- Combine https://github.com/rthalley/dnspython with https://github.com/prometheus/client_python into a Blackbox-style Prometheus exporter
- Export Metrics about query performance, responses, detailed failure info etc.
- Support all the protocols
- Support easy configuration from Prometheus scrape jobs

Seems easy enough

```
(venv) user@privat-dev:~/devel/dns_exporter/src$ pygount --
format=summary dns_exporter
```

Language	Files	%	Code	%	Comment	%
Python	9	90.0	1387	56.6	470	19.2
YAML	1	10.0	188	82.8	0	0.0
Sum	10	100.0	1575	58.8	470	17.6

```
(venv) user@privat-dev:~/devel/dns_exporter/src$
```

Installing

Install from pip in a venv:

```
pip install dns_exporter
```

There is also a Docker image so you can get your container on:

```
docker run -p 15353:15353 tykling/dns_exporter:latest
```

Running

Run the dns_exporter command to start the exporter and it should be ready to serve requests immediately:

```
$ dns_exporter
```

If you need more logging you can use -d or --debug:

```
$ dns_exporter -d
```

If you want to use a config file you can use -c or --config-file:

```
$ dns_exporter -c dns_exporter.yml
```

Configuration

- The exporter is configured on a per-lookup basis
- Available settings (most have defaults):
- *collect_ttl, collect_ttl_rr_value_length, edns, edns_bufsize, edns_do, edns_nsid, edns_pad, family, ip, protocol, proxy, query_class, query_name, query_type, recursion_desired, server, timeout, valid_rcodes, validate_additional_rrs, validate_answer_rrs, validate_authority_rrs, validate_response_flags, verify_certificate, verify_certificate_path*

Configuration

- Settings with no defaults:
  - server
  - query_name
- Default protocol is udp, other possibilities are:
  - tcp
  - udptcp
  - DoT
  - DoH
  - DoQ

Configuration File

- Reusable settings can be defined in modules and loaded in a config file when the exporter is started
- Config is a yaml file
- Example config installed with the package (from unit tests)
- Using a config can greatly simplify Prometheus scrape configs

Configuration File

```
tcp:
  protocol: "tcp"

tcpv4:
  protocol: "tcp"
  family: "ipv4"

ipv6:
  family: "ipv6"
```

Metrics

- The /query endpoint is used to do DNS queries and return metrics about that one lookup. These are reset with every scrape, no history.
- The /metrics endpoint returns internal exporter metrics with details about scrapes, failures, etc.

Per-scrape Metrics

- dnsexp_query_time_seconds (Gauge)
  - Labels: server, ip, port, protocol, family, proxy, query_name, query_type, transport, opcode, rcode, flags, answer, authority, additional, nsid
- dnsexp_dns_response_rr_ttl_seconds (Gauge)
  - Labels: Same as above plus rr_section, rr_name, rr_type, rr_value
- dnsexp_dns_query_success (Gauge)

Per-scrape Metrics

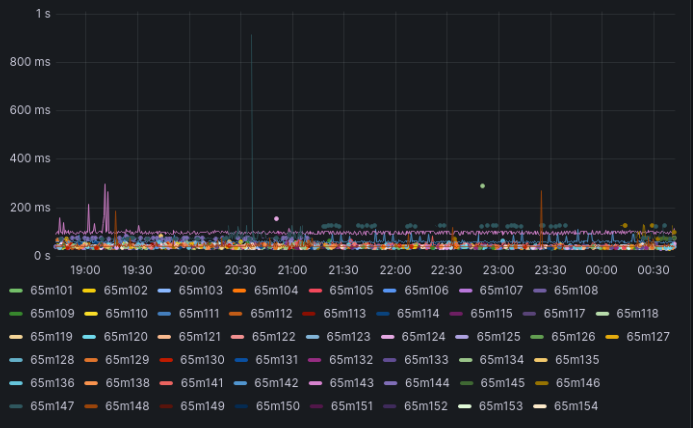
```
dnsexp_dns_query_time_seconds{additional="0",answer="1",authority="0",family="ipv4",flags="QR RA RD",ip="8.8.8.8",nsid="gpdns-ham",opcode="QUERY",port="53",protocol="udp",proxy="none",query_name="google.com",query_type="A",rcode="NOERROR",server="udp://dns.google:53",transport="UDP"} 0.013687849044799805

dnsexp_dns_response_rr_ttl_seconds{additional="0",answer="1",authority="0",family="ipv4",flags="QR RA RD",ip="8.8.8.8",nsid="gpdns-ham",opcode="QUERY",port="53",protocol="udp",proxy="none",query_name="google.com",query_type="A",rcode="NOERROR",rr_name="google.com.",rr_section="answer",rr_type="A",rr_value="172.217.19.78",server="udp://dns.google:53",transport="UDP"} 125.0

dnsexp_dns_query_success 1.0

up 1.0
```

Internal Metrics

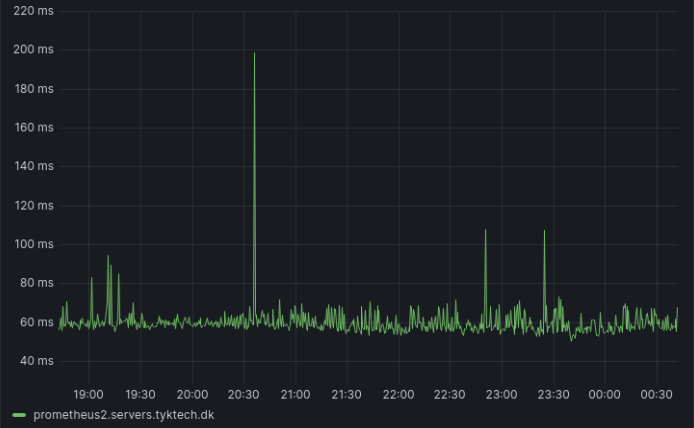
- dnsexp_build_version (Info)
- dnsexp_http_requests_total (Counter)
- dnsexp_http_responses_total (Counter)
- dnsexp_dns_queries_total (Counter)
- dnsexp_dns_responsetime_seconds (Histogram)
  - Buckets: .005, .01, .025, .05, .075, .1, .25, .5, .75, 1.0, 2.5, 5.0, 7.5, 10.0, INF
- dnsexp_scrape_failures_total (Counter)
  - Labels: Same as the per-scrape metrics, plus reason

DNS Response Time

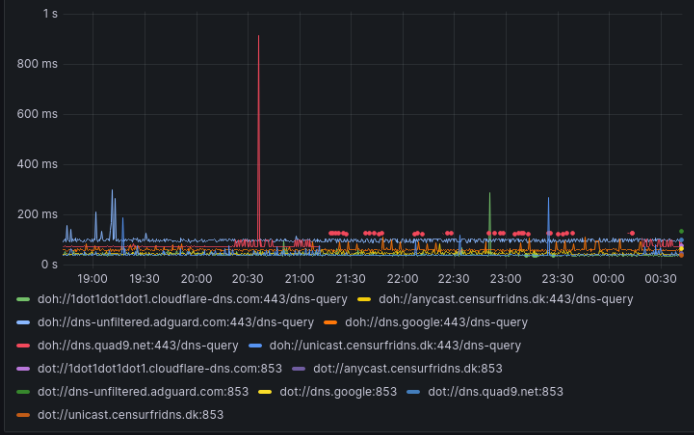
DNS Response Time by NSID



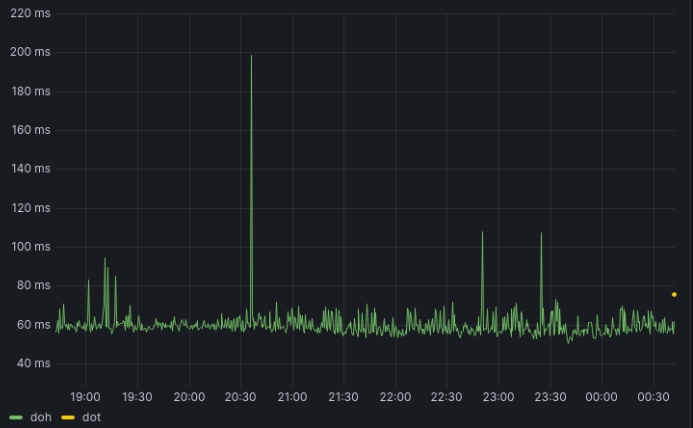
DNS Response Time by monitor



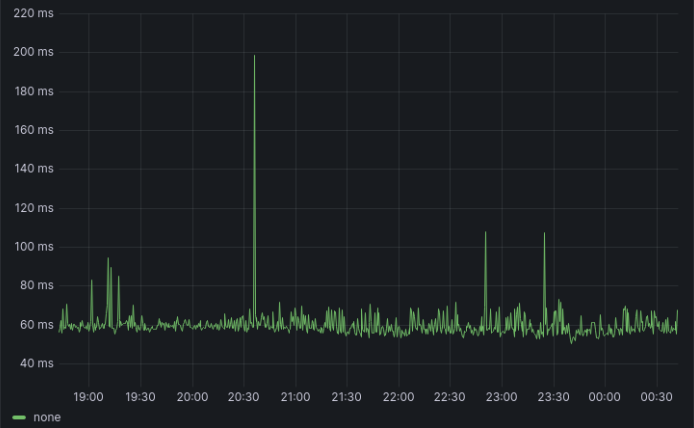
DNS Response Time by server



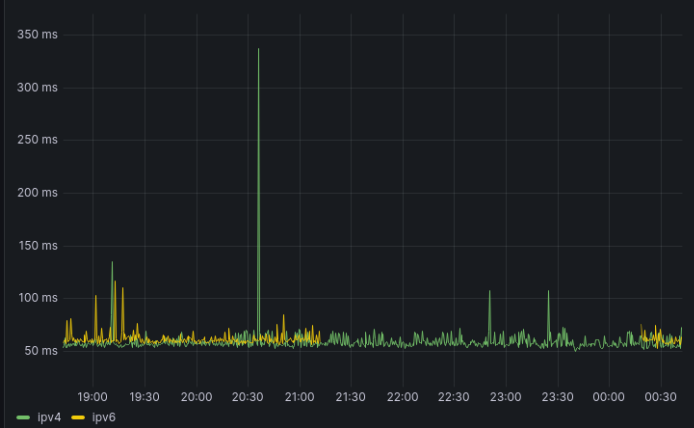
DNS Response Time by protocol



DNS Response Time by proxy



DNS Response Time by family

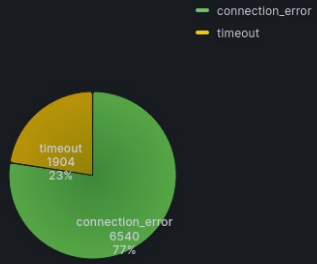


DNS Query Errors

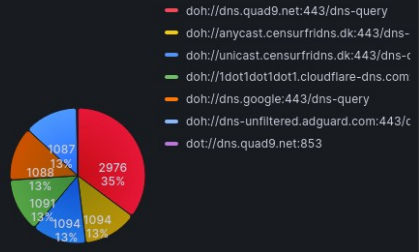
DNS Query Errors

8444

DNS Query Errors by Reason

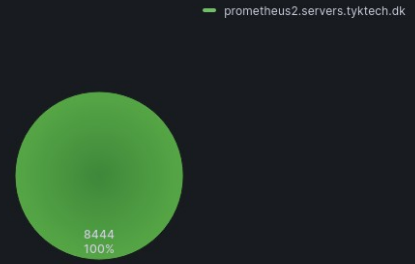


DNS Query Errors by Server

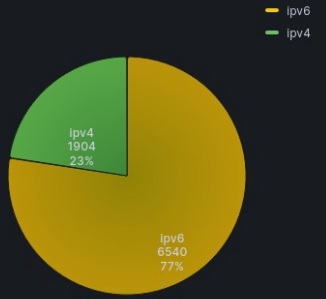


DNS Query Errors by Protocol

DNS Query Errors by Monitor



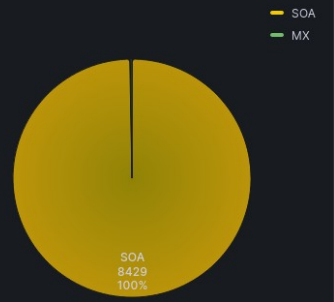
DNS Query Errors by IP Family



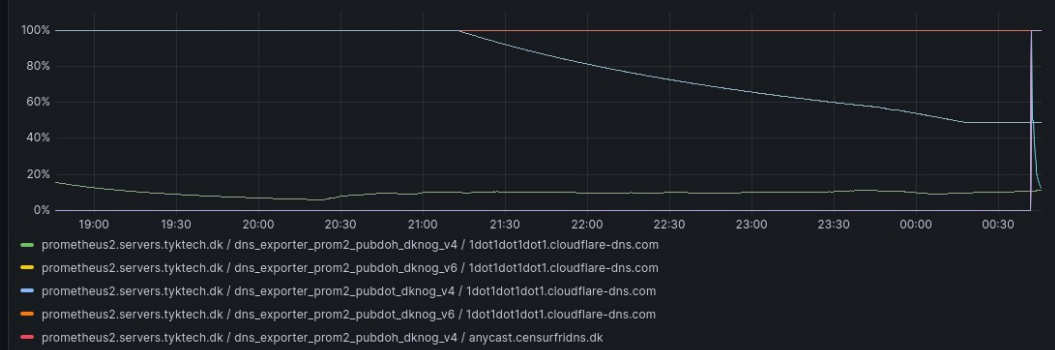
DNS Query Errors by IP Address

DNS Query Errors by Query Name

DNS Query Errors by Query Type



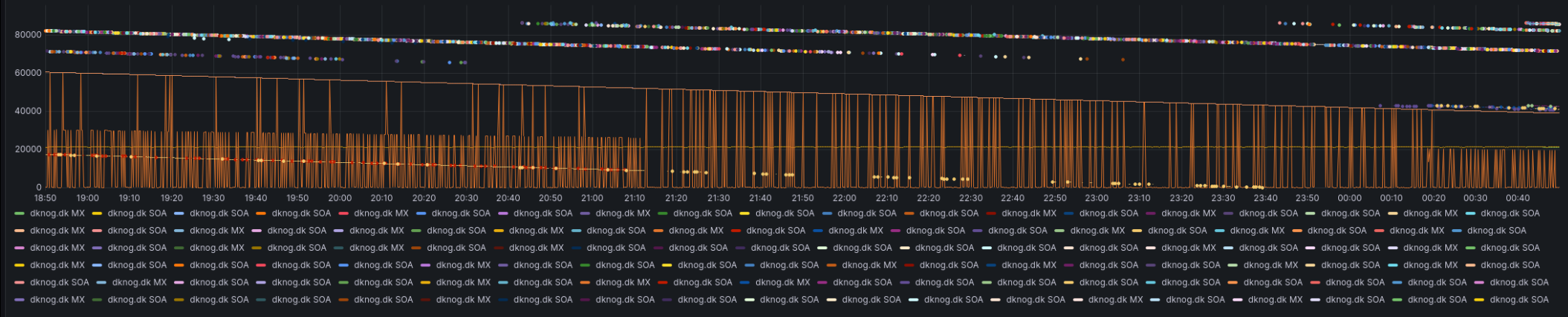
DNS Query Success Rate



Monitored Names

monitor	query_name	server	ip	flags	nsid	query_type	rr_name	rr_section	rr_type	rr_value	TTL
prometheus2.servers.t...	dknog.dk	doh://dns.quad9.net:4...	2620:fe::9	QR RA RD	res730.arn.rrdns.pch.net	SOA	dknog.dk.	answer	SOA	nordic1.dnsnode.net. e...	11.5 hours
prometheus2.servers.t...	dknog.dk	doh://unicast.censurfri...	2a01:3a0:53:53::	QR RA RD	unicast2.servers.cens...	SOA	dknog.dk.	answer	SOA	nordic1.dnsnode.net. e...	10.9 hours
prometheus2.servers.t...	dknog.dk	dot://1dot1dot1dot1.clo...	1.1.1.1	QR RA RD	65m130	MX	dknog.dk.	answer	MX	1 aspmx.l.google.com.	23.9 hours
prometheus2.servers.t...	dknog.dk	dot://1dot1dot1dot1.clo...	1.1.1.1	QR RA RD	65m130	MX	dknog.dk.	answer	MX	10 aspmx2.googlemail....	23.9 hours
prometheus2.servers.t...	dknog.dk	dot://1dot1dot1dot1.clo...	1.1.1.1	QR RA RD	65m130	MX	dknog.dk.	answer	MX	10 aspmx3.googlemail....	23.9 hours
prometheus2.servers.t...	dknog.dk	dot://1dot1dot1dot1.clo...	1.1.1.1	QR RA RD	65m130	MX	dknog.dk.	answer	MX	5 alt1.aspmx.l.google.c...	23.9 hours
prometheus2.servers.t...	dknog.dk	dot://1dot1dot1dot1.clo...	1.1.1.1	QR RA RD	65m130	MX	dknog.dk.	answer	MX	5 alt2.aspmx.l.google....	23.9 hours

DNS Response RR TTL



Finding Bugs

In own code
In dependencies
In DNS servers

dns.query.quic() cert validation with custom verify path broken? #1061

Open **tykling** opened this issue last week · 2 comments

tykling commented last week — Contributor

Describe the bug
Using `verify=/path/to/ca.pem` with `dns.query.quic()` doesn't seem to reject invalid certificates. Passing a selfsigned cert to verify= does not prevent a lookup from working, as demonstrated below.

Is it me doing something wrong? Thanks!

To Reproduce

```
(venv) user@privat-dev:~/devel/dns_exporter/src$ pip freeze | grep dnspython
dnspython==2.6.1
(venv) user@privat-dev:~/devel/dns_exporter/src$ python
Python 3.10.13 (main, Nov 15 2023, 13:09:29) [GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import dns.message, dns.name, dns.query
>>> q = dns.message.make_query(dns.name.from_text("example.com"), "A")
>>> dns.query.quic(q, "94.140.14.140", port=853, verify=True)
<DNS message, ID 0>
>>> dns.query.quic(q, "94.140.14.140", port=853, verify="/home/user/devel/dns_exporter/src/tests/certificates/
<DNS message, ID 0>
>>>
(venv) user@privat-dev:~/devel/dns_exporter/src$
```

The file `/home/user/devel/dns_exporter/src/tests/certificates/test.crt` contains a self-signed certificate.

I haven't got a DoQ server with an invalid certificate handy so I can't test which of these three scenarios is true, but I guess either:

1. no certificates are validated by `dns.query.quic()`
2. or more likely that the custom CA logic is a no-op
3. or that the custom CA logic doesn't remove system CAs when adding the custom ones

Context (please complete the following information):
see above

ps. I cut away a couple of aioquic cryptography deprecation warnings for clarity, they are not relevant to this

Thanks! :)

tykling commented last week • edited — Contributor Author

btw `94.140.14.140` is one of adguards doq servers and they do have IP:94.140.14.140 as a SAN so the certificate should (and does) validate with normal system CAs.

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
No branches or pull requests

2 participants

rthalley commented last week • edited

Owner

I can testify that if you give it bad certificates validation fails :).

The answer appears to be number 3 on your list, as aioquic always loads certifi and then also loads anything else. Dnspython will have passed the verify string into `cafile` in this aioquic TLS code:

```
    # load CAs
    store = crypto.X509Store()
    store.load_locations(certifi.where())
    if cadata is not None:
        for cert in load_pem_x509_certificates(cadata):
            store.add_cert(crypto.X509.from_cryptography(cert))

    if cafile is not None or capath is not None:
        store.load_locations(cafile, capath)
```

This was a bit surprising to me too as my expectation was like yours that if you specify a CA then it should be the only source. I'll ask Jeremy.

👍 1

rthalley mentioned this issue last week

Specifying cafile loads the CA file in addition to certifi and not instead of certifi
aiortc/aioquic#476

Open

tykling mentioned this issue last week

Custom CA support for QUIC does not remove system CAs tykling/dns_exporter#95

Open

add an overridable socket_factory to dns.quic._sync for #1059 #1060

Merged **rthalley** merged 1 commit into `rthalley:main` from `tykling:add_quic_socket_factory` 2 weeks ago

Conversation 1 | Commits 1 | Checks 9 | Files changed 1

tykling commented 2 weeks ago • edited — Contributor

#1059

`add an overridable socket_factory to dns.quic._sync for #1059` ✓ `af623b1`

codecov-commenter commented 2 weeks ago • edited

Codecov Report

All modified and coverable lines are covered by tests ✅

> Comparison is base (`a977e61`) 94.09% compared to head (`af623b1`) 94.10%.

❗ Your organization needs to install the Codecov GitHub app to enable full functionality.

▸ Additional details and impacted files

View full report in Codecov by Sentry.
Have feedback on the report? Share it here.

rthalley merged commit `8d535b9` into `rthalley:main` 2 weeks ago — View details

9 checks passed

tykling added a commit to tykling/dns_exporter that referenced this pull request 2 weeks ago

`add more proxy tests, disable proxy for DoQ until rthalley/dnspython#…` ✓ `5305ea2`

tykling mentioned this pull request last week

Add proxy support for QUIC tykling/dns_exporter#96 — Open

Reviewers
No reviews

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging th
these issues.
None yet

3 participants

DNSDist 1.9.0 DoH picks alpn http/1.1 over h2 when http/1.1 is listed first #13850

Closed | 2 tasks done | **tykling** opened this issue 4 days ago · 1 comment · Fixed by #13851

tykling commented 4 days ago • edited — Contributor

- [x] This is not a support question, I have read about opensource and will send support questions to the IRC channel, Github Discussions or the mailing list.
- [x] I have read and understood the 'out in the open' support policy

- Program: dnsdist 1.9.0
- Issue type: Bug report

Short description

`dnsdist 1.9.0` picks `http/1.1` over `h2` when both are offered in `alpn`, where `1.8.3` picks `h2`.

Environment

- Operating system: FreeBSD 13.2
- Software version: dnsdist 1.9.0
- Software source: DoH Client is `dnspython==2.6.1` which uses `httpx==0.26.0` which in turn uses `httpcore==1.0.2` which always adds http/1.1 to alpn. The same client works with http/2 on dnsdist 1.8.3.

Steps to reproduce

```
[tykling@irc2 ~]$ python3.9 -m venv venv
[tykling@irc2 ~]$ source venv/bin/activate
(venv) [tykling@irc2 ~]$ pip install dnspython[doh]
<snip>
Successfully installed anyio-4.3.0 certifi-2024.2.2 exceptiongroup-1.2.0 h11-0.14.0 h2-4.1.0 hpack-4.0.0 httpc
(venv) [tykling@irc2 ~]$ python
Python 3.9.16 (main, Dec 19 2022, 23:38:01)
[Clang 13.0.0 (git@github.com:llvm/llvm-project.git llvmorg-13.0.0-0-gd7b669b3a on freebsd13
Type "help", "copyright", "credits" or "license" for more information.
>>> import dns.message, dns.name, dns.query
>>> q = dns.message.make_query(dns.name.from_text("example.com"), "A")
>>> dns.query.https(q, "https://deic-lgb.anycast.uncensoreddns.org/dns-query") # this server runs dnsdist 1.9.0
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/home/tykling/venv/lib/python3.9/site-packages/dns/query.py", line 489, in https
    raise ValueError(
ValueError: https://deic-lgb.anycast.uncensoreddns.org/dns-query responded with status code 400
Response body: b'<html><body>This server implements RFC 8484 - DNS Queries over HTTP, and requires HTTP/2 in ac
>>> dns.query.https(q, "https://deic-ore.anycast.uncensoreddns.org/dns-query") # this server runs dnsdist 1.8.1
<DNS message, ID 46966>
>>>
```

Assignees
No one assigned

Labels
defect dnsdist

Projects
None yet

Milestone
dnsdist-1.9.x

Development
Successfully merging a pull request may close this issue.
tcpiohandler: Use server prefe
dwfreed/pdns

3 participants

send content-length headers for static content #100

Merged **tykling** merged 1 commit into `tykling:main` from `jcodybaker:send-content-length` yesterday

Conversation 2 | Commits 1 | Checks 0 | Files changed 1

jcodybaker commented yesterday — Contributor

I was using the / endpoint for health-checks in kubernetes. After a short while I noticed LOTs of connections stuck in timewait and the server was unresponsive via curl. I noticed this reference which makes sense prometheus/client_python#299 . Essentially the response either needs to use a chunked encoding, send a content length, or explicitly close the connection after sending the response (http/1.0 style).

Thanks for putting this tool together. It's exactly what I needed.

 `send content-length headers for static content` Verified `dc95f0c`

tykling approved these changes yesterday — View reviewed changes

tykling left a comment — Owner

good catch, thanks!

 **tykling** merged commit `400fe5d` into `tykling:main` yesterday

tykling commented yesterday — Owner

and I am happy to hear you find the tool useful!

Links

https://github.com/tykling/dns_exporter

https://dns-exporter.readthedocs.io/latest/

https://grafana.com/grafana/dashboards/20617

https://hub.docker.com/r/tykling/dns_exporter

https://pypi.org/project/dns-exporter/

https://dnsgrafana.tyktech.dk/d/UnXfnkh4z/dns-exporter

Welcome to UncensoredDNS

UncensoredDNS is the name of a DNS service which consists of two uncensored DNS servers. The servers are available for use by anyone, free of charge.

This service is run by Thomas Steen Rasmussen, born 1979. I am a system architect and developer in a Danish company, and I also teach and consult in my spare time. I run UncensoredDNS as a private individual, with my own money.

You can read more using the menu above, or if you just want the DNS server info can get it below.

DNS servers

anycast.uncensoreddns.org

Anycast from multiple locations.

DNS-over-TLS anycast.uncensoreddns.org:853

DNS-over-HTTPS https://anycast.uncensoreddns.org/dns-query

91.239.100.100
2001:67c:28a4::

unicast.uncensoreddns.org

This node is hosted at AS9167 in Copenhagen, Denmark.

DNS-over-TLS unicast.uncensoreddns.org:853

DNS-over-HTTPS https://unicast.uncensoreddns.org/dns-query

89.233.43.71
2a01:3a0:53:53::

BORNHACK

July 17. to 24. 2024 Funen, Denmark

BornHack is a 7 day **outdoor tent camp** where hackers, makers and people with an interest in technology or security come together to celebrate technology, socialise, learn and **have fun**.

BornHack 2024 will be the ninth BornHack. It will take place from **Wednesday the 17th of July to Wednesday the 24th of July 2024** at our venue on the Danish island of Funen.

The End!

Questions?