Knot DNS Resolver*

Marek Vavrusa
CZ.NIC

*work in progress
knot-resolver.rtfd.org

if you feel like reading...
Library services

Resolution
Modules and layers
Cache
*Reputation
“Name resolution” is data transformation in disguise
“Name resolution” is data transformation in disguise
Modules and layers

- **Layer** is an independent SM
- Mix and match functionality
- Less active code, less attack surface area
- Iterator, record cache, packet cache

*artistic representation of the layers*
Showcase: “iterator” layer

• Drives the query resolution, cooperates with caching layers

• Does best-effort QNAME minimisation
  • meaning it stops minimising when it encounters a zone cut, but why?
  • broken CDNs, PTR records with many labels, …
• **Generic storage backends** (default - LMDB)

• **Tagged resources** (record, packet, secret …)

• **Persistent with default backend**

• **Replaceable on runtime**
NS reputation

“Tracks NS latency and crimes against the DNS”

(Not persistent now, work in progress)
Resolver daemon

- Written in C and Lua
- Built around the libuv library
- Dynamic configuration, CLI, extensions, … even layers in Lua
Dynamic what?

It’s really Lua behind, but you can configure it declaratively

```lua
-- static example
cache.size = 10*MB
modules = {
    'hints', 'cachectl'
}
net = {
    '127.0.0.1'
}
```
Dynamic what?

Listen on all `eth0` addresses.

```plaintext
net.listen(net.eth0)
event.recurrent(30*sec, function()
  cachectl.prune()
end)
```

Prune cache every 30 seconds.
Resolver modules 101

- Add processing layers
  (...OpenResty of DNS)
- Script *anything*
- Subscribe / publish data
- C, Lua or Go*
Batteries included, but removable

• Static hints (C, processing layer)

• Etcd (Lua, updates configuration from peers)

• Cache control (C, pruning/purging cache)

• Memcached (C, new cache storage option)
Recap

• Resolver library with a “state-machine-esque” API
• Scriptable daemon with dynamic configuration
• Bunch of modules in C/Go/Lua
• Quarterly release plan, but things might break!
Demo

... not really
marek$ make

Dependencies
-------------
[yes] doxygen (doc)
[yes] libknot (lib)
[yes] lua (daemon)
[yes] libuv (daemon)
[yes] cmocka (tests/unit)
[yes] Python (tests/integration)
[no] GCCGO (modules/go)

make: Nothing to be done for `all'.
Dependences
------------
[yes] doxygen (doc)
[yes] libknot (lib)
[yes] lua (daemon)
[yes] libuv (daemon)
[yes] cmocka (tests/unit)
[yes] Python (tests/integration)
[no] GCCGO (modules/go)

make: Nothing to be done for `all'.

marek$ ./daemon/kressembled -a 127.0.0.1#6667
[system] started in interactive mode, type 'help()'
marek$ ./daemon/kresolved -a 127.0.0.1#6667
[system] started in interactive mode, type 'help()'
> net.list()
[127.0.0.1] => {
  [udp] => true
  [tcp] => true
  [port] => 6667
}
>
marek$ ./daemon/kresolved -a 127.0.0.1#6667
[system] started in interactive mode, type 'help()'
> net.list()
[127.0.0.1] => {
    [udp] => true
    [tcp] => true
    [port] => 6667
}
> modules = {'hints', 'cachectl'}
[hint] reading '/etc/hosts'
[hint] loaded 9 hints
marek$ ./daemon/kresolved -a 127.0.0.1#6667
[system] started in interactive mode, type 'help()'

> net.list()

[127.0.0.1] => {
    [udp] => true
    [tcp] => true
    [port] => 6667
}

> modules = {'hints', 'cachectl'}

[hint] reading '/etc/hosts'
[hint] loaded 9 hints

> modules.list()

[1] => iterate
[2] => rrcache
[3] => pktcache
[4] => hints
[5] => cachectl
marek$ kdig @127.0.0.1 -p 6667 A nic.cz.
;; ->>HEADER<<- opcode: QUERY; status: NOERROR; ...
;; Flags: qr rd ra; QUERY: 1; ANSWER: 1; ...

;; QUESTION SECTION:
;; nic.cz. IN A

;; ANSWER SECTION:
nic.cz. 1800 IN A 217.31.205.50

;; Received 40 B
;; Time 2015-05-13 16:34:43 CEST
;; From 127.0.0.1@6667(UDP) in 42.3 ms
marek$ ./daemon/kresolved -a 127.0.0.1#6667
[system] started in interactive mode, type 'help()'
> ...
[plan] plan 'nic.cz.' type 'A'
[resl] query 'A nic.cz.'
[resl] => querying ... cut: '.' m12n: 'cz.'
[iter] <= referral response, follow
[resl] => querying ... cut: 'cz.' m12n: 'nic.cz.'
[iter] <= rcode: NOERROR
marek$ ./daemon/kresolved -a 127.0.0.1#6667
[system] started in interactive mode, type 'help()'
> ...
> hints['badguy'] = '127.0.0.1'
marek$ ./daemon/kresolved -a 127.0.0.1#6667
[system] started in interactive mode, type 'help()'
> ...
> hints['badguy'] = '127.0.0.1'
> hints['badguy']
{ "result": [ 127.0.0.1 ] }
marek$ kdig @127.0.0.1 -p 6667 A badguy.

;; -->>HEADER<<-- opcode: QUERY; status: NOERROR ...
;; Flags: qr rd ra; QUERY: 1; ANSWER: 1; ...

;; QUESTION SECTION:
;; badguy. IN A

;; ANSWER SECTION:
badguy. 0 IN A 127.0.0.1

;; Received 40 B
;; Time 2015-05-13 16:08:00 CEST
;; From 127.0.0.1@6667(UDP) in 0.2 ms
marek$ ./daemon/kresolved -a 127.0.0.1#6667
[system] started in interactive mode, type 'help()'
> ...
> hints['badguy'] = '127.0.0.1'
> hints['badguy']
{ "result" : [ 127.0.0.1 ] }
>
[plan] plan 'badguy.' type 'A'
[resl] query 'A badguy.'
[hint] <= answered from hints
[iter] <= rcode: NOERROR
Q/A ?

github.com/CZ-NIC/knot-resolver

gitlab.labs.nic.cz/knot/resolver

travis-ci.org/CZ-NIC/knot-resolver

scan.coverity.com/projects/3912

knot-resolver.rtfd.org