



Deckard – DNS “test lab”

I’ve seen things you DNS people wouldn’t believe

Ondřej Surý • ondrej.sury@nic.cz • 01.04.2016



Deckard – a software “test lab”

- Developed during Knot DNS Resolver development
 - Simulate everything on runtime
 - Creates a controlled environment
- Wraps around syscalls and standard library functions, thus...
 - The tests are fast
 - Testing is done in seconds
- Thus could be integrated into development cycle
 - Continuous integration



Deckard – components

- `socket_wrapper`
 - Creates fake network environment
 - Application can bind to privileged ports
 - Communicate with mocked servers
- `libfaketime`
 - Change the flow of the time
- `Jinja2`
 - For creating configuration from templates
- `Dnspython`
 - For parsing and constructing DNS packets



Test scenarios

- Inspired by Unbound replay.h test cases
- Runs the production binary as a subprocess
- Redirects all network communication to UNIX sockets
- Declarative description of the environment
 - DNS server configuration
 - Network configuration
- Sequence of DNS messages
 - Queries to be asked
 - Answers to be given
 - Expectations about answers



Example – Knot Resolver J2 template

```
-- this is Knot Resolver configuration file
-- bind to fake address
net = { '{{SELF_ADDR}}' }
-- load some needed modules
modules = {'stats', 'policy', 'hints'}
-- Set K-root address to a fake internal address
hints.root({'k.root-servers.net'] = '{{ROOT_ADDR}}'})
-- Enable or disable QNAME minimization
option('NO_MINIMIZE', '{{NO_MINIMIZE}}')
option('ALLOW_LOCAL', false)
-- Add our trust anchor
trust_anchors.add('{{TRUST_ANCHOR}}')
-- Enable verbose logging
verbose(true)
```



Example – Scenario Basics

- RANGE m n – RANGE_END
 - Define a set of queries/answers used in STEPS m-n
- ENTRY_BEGIN – ENTRY_END
 - A description of DNS message
- MATCH
 - Which queries does trigger the answer?
 - all, opcode, qtype, qname, ...
- ADJUST
 - copy_id || copy_query
- REPLY <flag>|<rcode>
 - Set flags
 - Set RCODE
- SECTION <type> – END_SECTION
 - RR data to return in the DNS message
- STEP <n> <type>
 - QUERY | REPLY | CHECK_ANSWER | ...
- RAW
 - Encoded binary content



Example – Basic Iteration Test

```
; query-minimization: on/off  
; stub-addr: ipaddr ; matches {{ROOT_ADDR}}  
; trust-anchor: <string>  
; val-override-timestamp: <timestamp>  
; val-override-date: YYYYMMDDHHMMSS  
; [...]  
CONFIG_END
```

```
SCENARIO_BEGIN <scenarion description>
```

```
; <ranges_definition>  
; <steps_definition>
```

```
SCENARIO_END
```



Ranges (1) – k.root-servers.net

RANGE_BEGIN 0 100
ADDRESS 193.0.14.129
ENTRY_BEGIN
MATCH opcode qtype qname
ADJUST copy_id
REPLY QR NOERROR
SECTION QUESTION
. IN NS
SECTION ANSWER
. IN NS K.ROOT-SERVERS.NET.
SECTION ADDITIONAL
K.ROOT-SERVERS.NET. IN A \
193.0.14.129
ENTRY_END

ENTRY_BEGIN
MATCH opcode qtype qname
ADJUST copy_id
REPLY QR NOERROR
SECTION QUESTION
www.example.com. IN A
SECTION AUTHORITY
com. IN NS a.gtld-servers.net.
SECTION ADDITIONAL
a.gtld-servers.net. IN A \
192.5.6.30
ENTRY_END

RANGE_END



Ranges (2) – a.gtld-servers.net

RANGE_BEGIN 0 100

ADDRESS 192.5.6.30

ENTRY_BEGIN

MATCH opcode qtype qname

ADJUST copy_id

REPLY QR NOERROR

SECTION QUESTION

. IN NS

SECTION ANSWER

. IN NS a.gtld-servers.net.

SECTION ADDITIONAL

a.gtld-servers.net. IN A \

192.5.6.30

ENTRY_END

ENTRY_BEGIN

MATCH opcode qtype qname

ADJUST copy_id

REPLY QR NOERROR

SECTION QUESTION

www.example.com. IN A

SECTION AUTHORITY

example.com. IN NS \

ns.example.com.

SECTION ADDITIONAL

ns.example.com. IN A \

192.0.2.1

ENTRY_END

RANGE_END



Ranges (3) – ns.example.com

RANGE_BEGIN 0 100

ADDRESS 192.0.2.1

ENTRY_BEGIN

MATCH opcode qtype qname

ADJUST copy_id

REPLY QR NOERROR

SECTION QUESTION

. IN NS

SECTION ANSWER

. IN NS ns.example.com.

SECTION ADDITIONAL

ns.example.com. IN A \

192.0.2.1

ENTRY_END

ENTRY_BEGIN

MATCH opcode qtype qname

ADJUST copy_id

REPLY QR NOERROR

SECTION QUESTION

www.example.com. IN A

SECTION ANSWER

www.example.com. IN A \

203.0.113.1

SECTION AUTHORITY

example.com. IN NS \

ns.example.com

SECTION ADDITIONAL

ns.example.com. IN A \

192.0.2.1

ENTRY_END

RANGE_END



Steps

STEP 1 QUERY

ENTRY_BEGIN

REPLY RD

SECTION QUESTION

www.example.com. IN A

ENTRY_END

; recursion happens here

STEP 10 CHECK_ANSWER

ENTRY_BEGIN

MATCH all

REPLY QR RD RA NOERROR

SECTION QUESTION

www.example.com. IN A

SECTION ANSWER

www.example.com. IN A 203.0.113.1

;SECTION AUTHORITY

;example.com. IN NS

ns.example.com.

;SECTION ADDITIONAL

;ns.example.com. IN A 192.0.2.1

ENTRY_END



Deckard – further development

- Free software, open-source
 - <https://gitlab.labs.nic.cz/knot/deckard>
- Scenario guide and more scenarios:
 - https://gitlab.labs.nic.cz/knot/deckard/blob/master/SCENARIO_GUIDE.rst
 - <https://gitlab.labs.nic.cz/knot/deckard/tree/master/sets>
- More complicated scenario example:
 - https://gitlab.labs.nic.cz/knot/deckard/blob/master/SCENARIO_EXAMPLE.rst
- You are welcome to participate
 - More test cases – for resolvers, authoritative DNS, and even DNS tools
 - More servers to test – Jinja2 templates for the configuration
 - More (and regular) testing
 - Perhaps write an I-D?



Thanks!

