



Framework for DNS traffic pattern generation

Sebastian Castro
sebastian@nzrs.net.nz

NZRS



Motivation

- What's an acceptable query load for a server before going over certain response time?
- Does DNSSEC change the picture?
 - Positive/Negative answer have different costs
- Heavily inspired by the RZAIA report
- When replying old traffic is not enough

Two components

- Pattern generator
 - Create DNS traffic patterns based on statistical properties
- Pattern player
 - Take a pattern and send it to a nameserver
 - You can adjust what to get from the test, how fast to send the queries, etc.

Pattern generator

- Written in Ruby
- Two sub-components
 - NXDOMAIN generator
 - query stream generator
- Dependencies
 - JSON
 - GNU Scientific Library (GSL)
- Uses Probability Distribution Functions to generate a pattern

Pattern generator

- Parameters:
 - Number of queries
 - Hit Ratio
 - EDNS Ratio
 - DO bit ratio
 - Recursion ratio
 - EDNS Buffer size distribution file
 - Query type distribution file
- Output: JSON file

Pattern Player

- Modified queryperf to:
 - Support reading the input JSON file (instead of plain text, one query per line input)
 - Retry queries with truncated responses
 - Track query and response sizes.
- Dependencies
 - jansson (JSON library)
 - Idns (for response parsing)

Walk-trough (1)

- One or more zone files as input
 - make ZONEFILE=co.nz.axfr hit-names.txt
- Generate the miss name file
 - make miss-names.txt
 - Relies in a “distribution file” with the frequency of suffixes
- Probability distribution for query types
- Probability distribution for EDNS buffer size

```
ruby miss-name-generator.rb -n 5 -l 12 -e  
hit-names.txt  
t5bc09uk6lp4.co.nz.  
ggr00emy0hd2.co.nz.  
e3xgyodfsud3.net.nz.  
coxnsja9sp69.co.nz.  
whvoyw8v8qdv.net.nz.
```

sld-distrib.txt file

```
0.800  co.nz  
0.200  net.nz
```

Walk-trough (2)

- Query type probability distribution sample
- EDNS buffer size probability distribution sample

```
0.6284 A
0.0201 NS
0.0024 CNAME
0.0050 SOA
0.0111 PTR
0.1657 MX
0.0079 TXT
0.1432 AAAA
0.0039 SRV
0.0003 SPF
0.0086 A6
```

```
0.0099 512
0.0250 1024
0.0221 2048
0.9379 4096
```


Walk-trough (3)

- Ratios
 - Hit ratio
 - EDNS ratio
 - DO bit ratio
 - Recursive ratio
- Generate list of queries
 - make query-list.dat

```
./querygen.rb --num-queries=100000 --edns-ratio=0.4750  
--do-bit-ratio=0.4439 --recursive-ratio=0.28 --hit-  
ratio=0.93 --edns-buffer-size-file=edns-buffer-size.dat  
--qtype-distrib-file=qtype.dat --hit-names-file=hit-  
names.txt --miss-names-file=miss-names.txt
```

Walk-trough (4)

- Send the queries to a server
 - `queryperf -d query-list.dat -s 192.168.22.152 -1`

Statistics:

```
AA bit responses:      4017
TC bit responses:      10
TCP retries sent:      0
Bytes sent:            2084833
Bytes received:        9260572
Avg query size:        41.70 bytes
Avg response size:     185.21 bytes

RTT max:               0.013115 sec
RTT min:               0.000122 sec
RTT average:           0.000786 sec
RTT std deviation:     0.000596 sec
RTT out of range:     0 queries

Returned NOERROR:     45983 queries
  Avg resp size:       181.438 bytes
Returned NXDOMAIN:    4017 queries
  Avg resp size:       228.411 bytes

Queries per second:    23339.042063 qps
```

Real tests

- Two patterns
 - “Normal” traffic pattern
 - “MX Burst”
- Three nameservers implementation
 - BIND, NSD, Knot
- One signed zone (co.nz)
 - NSEC3, Opt-out

Real test: Patterns

- Normal
 - Hit Ratio: 0.92
 - Recursive ratio: 0.07
 - EDNS ratio: 0.56
 - DO ratio: 0.52
 - Qtypes: 0.63 A, 0.17 MX, 0.14 AAAA + others
 - EDNS buf size: mostly 4096 bytes
- MX Bursts
 - Hit Ratio: 0.45
 - Recursive Ratio: 0.65
 - EDNS ratio: 0.20
 - Do Ratio: 0.19
 - Qtypes: 0.65 MX, 0.28 A, 0.05 AAAA + others
 - EDNS buf size: mostly 4096 bytes

Results

Software	Normal			MX		
	QPS	Avg. Resp. Time [ms]	Avg. Resp. Size [byte]	QPS	Avg. Resp. Time [ms]	Avg. Resp. Size [byte]
BIND 9.7	23736	0.786	185.21	30311	0.575	112.10
NSD 3	32669	0.523	201.81	33927	0.495	122.64

- “MX” test, with more NXDOMAIN responses, produces shorter answers?
- Same stream of queries, NSD returns more data per response?

Results

Software	Normal			MX		
	QPS	Avg. Resp. Time [ms]	Avg. Resp. Size [byte]	QPS	Avg. Resp. Time [ms]	Avg. Resp. Size [byte]
BIND 9.7	23736	0.786	185.21	30311	0.575	112.10
NSD 3	32669	0.523	201.81	33927	0.495	122.64

- “MX” test, with more NXDOMAIN responses, produces shorter answers?
- Same stream of queries, NSD returns more data per response?

Results

- The MX pattern has a low number of DNSSEC-OK queries.
- What if we change the pattern?
 - Set EDNS ratio to 0.80, DO bit ratio to 0.79

Software	MX (1)			MX (2)		
	QPS	Avg. Resp. Time [ms]	Avg. Resp. Size [byte]	QPS	Avg. Resp. Time [ms]	Avg. Resp. Size [byte]
BIND 9.7	30311	0.575	112.10	21853	0.839	323.51
NSD 3	33927	0.495	122.64	29013	0.610	333.80
Knot 1.0.1						

Limitations

- If a zone contains a wildcard, the hit ratio won't work
- Heavily designed for delegation-centric zone testing
- The input file has to reside in memory
 - Restrictions in the input size

Conclusions

- Plenty of possibilities for testing
- Bringing some of the features seen in network testing to the DNS
- Code will be available at
github.com/NZRS/dns-traffic

