dnsdist: high-performance, DoS and abuse-aware DNS loadbalancer

Nico Cartron, Remi Gacogne

FRnOG, March 27th 2017

POWERDNS AN OX COMPANY

Presentation

- Nico Cartron
  - Senior Sales Engineering @ PowerDNS / OX
- Remi Gacogne
  - Senior Software Engineer @ PowerDNS / OX

Open-Xchange: An Integrated Stack

Adding to the family of robust products

In 2015, Dovecot and PowerDNS merged with Open-Xchange to become the leading Open Source powerhouse of messaging & collaboration services

DOVECOT
AN OX COMPANY

- 4M mail server installations globally
- 71,67% worldwide market share
- Highly scalable and cost efficient
- Fully secure

POWERDNS AN OX COMPANY

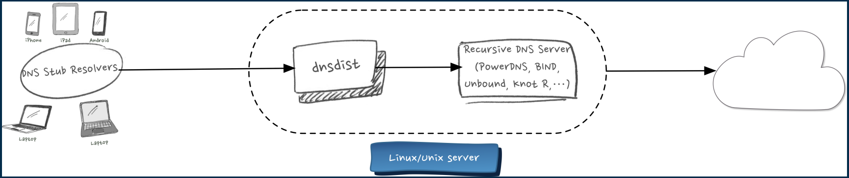
- EU market leader (30%)
- DNSSEC >75% of hosted domains
- Excellent scalability
- Best in class DoS support

dnsdist – History and Origins

```
dnsdist listen-ip dest-ip-1 dest-ip-2
```

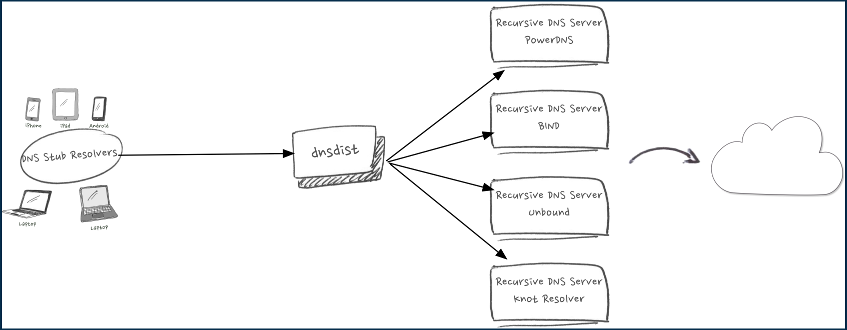
- Most load balancers know about HTTP(S), IMAP etc.
- DNS can’t be handled as “a weird kind of web”
- Observation: A busy nameserver is a happy nameserver
- “concentrating load balancer”

dnsdist – Use cases



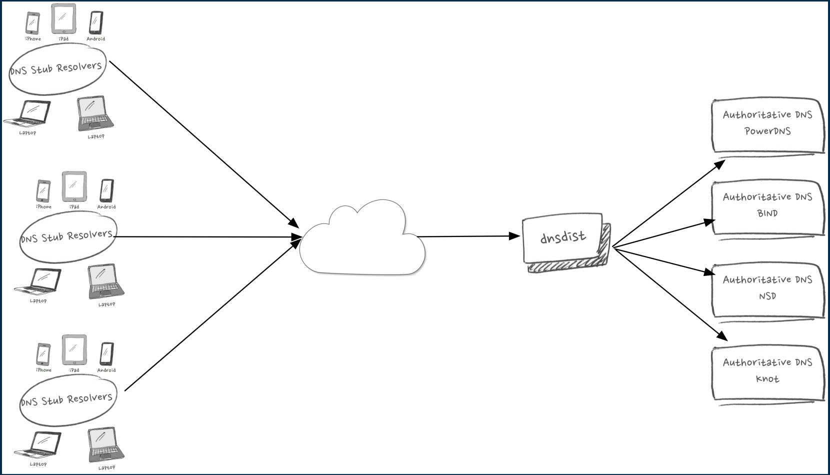
On the same host, gives statistics and saves requests history

dnsdist – Use cases II


iPhone
iPad
Android
DNS Stub Resolvers
Laptop
Laptop
dnsdist
Recursive DNS Server PowerDNS
Recursive DNS Server BIND
Recursive DNS Server Unbound
Recursive DNS Server Knot Resolver

In front of Recursive servers, protects, balances and filters traffic

dnsdist – Use cases III



In front of Authoritative servers, protects, balances and filters traffic

dnsdist – Features

- Configuration at runtime via the console (local / remote)
- Product core and rules written in C++
- Fully manageable using Lua (config, rules, LB policy...)
- Blazing fast in-memory packet cache
- Very low memory usage: a few MB without caching
- Low CPU usage: several hundred thousand QPS on a single core

dnsdist – LB policies

- Least Outstanding (default)
- First Available
- Weighted hashed
- Round Robin
- Weighted random
- Custom

dnsdist – Rules

Based on the source, the content, the time of the day...

- Alter the query content (flags, EDNS Client Subnet, ...)
- Route the query to a specific servers pool (“abuse”)
- Drop the query
- Delay or Spoof a response
- Detect and mitigate DoS, infected clients (userspace / kernel via eBPF)

dnsdist - Default configuration

```
dnsdist -l 192.0.2.100:53 192.0.2.1 192.0.2.2
```

- Listen on port 53
- Accept queries from RFC 1918 addresses by default
- Distribute queries to 192.0.2.1 and 192.0.2.2
- Use a sensible loadbalancing policy (“leastOutstanding”)

dnsdist – Simple configuration

```
setLocal('192.0.2.100:53')
setACL('192.0.2.0/24')
newServer{address='192.0.2.1', qps=1000, order=1}
newServer{address='192.0.2.2', order=2}
setServerPolicy(firstAvailable)
```

dnsdist – Simple configuration II

```
# dnsdist -C simple.lua
Added downstream server 192.0.2.1:53
Added downstream server 192.0.2.2:53
Listening on 192.0.2.100:53
dnsdist 0.0.gf354a19 comes with ABSOLUTELY NO WARRANTY. This is free software, and you are welcome to redi
ACL allowing queries from: 192.0.2.0/24
Marking downstream 192.0.2.1:53 as 'up'
Marking downstream 192.0.2.2:53 as 'down'
> showServers()
#   Name  Address         State  Qps  Qlim Ord Wt Queries Drops Drate Lat Outstanding
0         192.0.2.1:53       up  0.0  1000   1  1       0     0   0.0 0.0           0
1         192.0.2.2:53     down  0.0     0   2  1       0     0   0.0 0.0           0
All                              0.0                    0     0

> getServer(1):setDown()
> showServers()
#   Name  Address         State  Qps  Qlim Ord Wt Queries Drops Drate  Lat Outstanding
0         192.0.2.1:53       up  0.0  1000   1  1      18     0   0.0  9.4           0
1         192.0.2.2:53     DOWN  0.0     0   2  1       0     0   0.0  0.0           0
All                              0.0                   18     0
```

dnsdist – Live traffic inspection

```
> showResponseLatency()
Average response latency: 0.582 msec
   msec
   0.10 .
   0.20 ****
   0.40 **********************************************************************
   0.80 ****
   1.60 .
   3.20
   6.40
  12.80
  25.60 *****
  51.20 ********
 102.40 *******
 204.80 *****
 409.60 ****
 819.20 *
1638.40 .
```

dnsdist – Live traffic inspection II

```
> topQueries(5)
   1  hehehey.ru.                                  2358 23.6%
   2  localhost.                                   2281 22.8%
   3  time.apple.com.                               537  5.4%
   4  service-personal.de.                          144  1.4%
   5  time.euro.apple.com.                          109  1.1%
   6  Rest                                         4571 45.7%

> topSlow(4)
   1  148.117.189.193.in-addr.arpa.                   3  2.4%
   2  _sipfederationtls._tcp.helpdesk.symphony.com.my.    2  1.6%
   3  eu2-scloud-proxy.ssp.samsungosp.com.            2  1.6%
   4  219.116.189.193.in-addr.arpa.                   2  1.6%
   5  Rest                                          114 92.7%

> topResponses(2, dnsdist.SERVFAIL)
   1  150.209.45.194.in-addr.arpa.                   31 22.1%
   2  praesenzen.datevstadt.de.                      15 10.7%
   3  Rest                                           94 67.1%
```

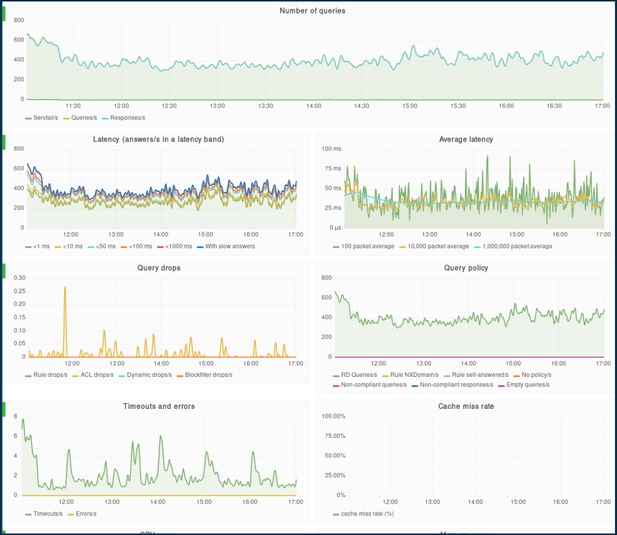
dnsdist – Live traffic inspection III

```
> grepq('ru', 2)
Time  Client              Server          ID    Name         Type Lat.  TC RD AA Rcode
-0.2  192.0.2.92:33846                    4905  hehehey.ru.  ANY           RD    Question
-0.2  192.0.2.92:33846    127.0.0.1:5300  4905  hehehey.ru.  ANY  0.2      RD    Non-Existent domain
-0.2  192.0.2.92:33846                    4907  hehehey.ru.  ANY           RD    Question
-0.2  192.0.2.92:33846    127.0.0.1:5300  4907  hehehey.ru.  ANY  0.3      RD    Non-Existent domain

> grepq({'apple.com.', "100ms"}, 5)
Time    Client             Server          ID    Name           Type Lat.   TC RD AA Rcode
-127.6  192.0.2.92:43583   127.0.0.1:5300  44987 cl4.apple.com. A    247.2     RD    No Error. 4 answers
```

dnsdist – Carbon export

Built-in export of metrics via Carbon (Graphite / Metronome)



dnsdist – Protocol Buffer

```
rl = newRemoteLogger("192.0.2.1:4242")
addAction(AllRule(), RemoteLogAction(rl))
addResponseAction(AllRule(), RemoteLogResponseAction(rl))
```

```
[2016-11-08 21:45:34.351969] Query of size 51: 127.0.0.1 -> 127.0.0.1 (UDP),
    id: 20205, uuid: 0225802a7e9446aa9e4e915102c28910
- Question: 1, 1, kernel.org.
[2016-11-08 21:45:36.61240] Response of size 87: 127.0.0.1 -> 127.0.0.1 (UDP),
    id: 20205, uuid: 0225802a7e9446aa9e4e915102c28910
- Question: 1, 1, kernel.org.
- Query time: 2016-11-08 21:45:34.352025
- Response Code: 0, RRs: 3
         - 1, 1, kernel.org., 600, 199.204.44.194
         - 1, 1, kernel.org., 600, 198.145.20.140
         - 1, 1, kernel.org., 600, 149.20.4.69
[2016-11-08 21:45:40.158478] Query of size 51: 127.0.0.1 -> 127.0.0.1 (UDP),
    id: 24445, uuid: 07939afeddfe4089a2d4fd56f5aca755
- Question: 1, 28, kernel.org.
[2016-11-08 21:45:40.303994] Response of size 95: 127.0.0.1 -> 127.0.0.1 (UDP),
    id: 24445, uuid: 07939afeddfe4089a2d4fd56f5aca755
- Question: 1, 28, kernel.org.
- Query time: 2016-11-08 21:45:40.158534
- Response Code: 0, RRs: 2
         - 1, 28, kernel.org., 600, 2001:4f8:1:10:0:1991:8:25
         - 1, 28, kernel.org., 600, 2620:3:c000:a:0:1991:8:25
```

dnsdist – API

```
$ http http://127.0.0.1:8084/api/v1/servers/localhost/statistics X-API-Key:secretapikey
[
    {
        "name": "queries",
        "type": "StatisticItem",
        "value": 2445
    },
    {
        "name": "responses",
        "type": "StatisticItem",
        "value": 2439
    },
    {
        "name": "servfail-responses",
        "type": "StatisticItem",
        "value": 0
    },
[...]
```

dnsdist – For a Few Rules More

Rules have Selectors with Actions

Selector: does this rule apply?
Actions: Do X if I match

Rules evaluated top-to-bottom, first match wins

dnsdist – Selectors

- Source or destination address
- Query features (QNAME, QTYPE, Flags)
- Number of entries in a packet sections
- Number of labels, length of the name
- Regular Expression (POSIX, RE2)
- Supports And, Or and Not

dnsdist – Actions

- Drop
- Route to Pool
- Truncate (TC=1)
- Return SERVFAIL, NOTIMP, REFUSED
- Return custom answer
- Delay response by n milliseconds
- Remove flags before passing to backend
- Add originating IP address in an EDNS Client Subnet option
- Log query to TCP/IP host via Protobuf

dnsdist – Examples

Let's say we are flooded by some CPE sending DNS queries in a loop:

```
addAction(MaxQPSIPRule(5, 24, 64), DropAction())
↪   -- 5 QPS, grouped by /24 on IPv4 and by /64 on IPv6
addAction("domain.targeted.example.", DelayAction(500))
↪   -- delay responses for this domain by 500ms
addAction("suspicious.example.", PoolAction("Abuse"))
↪   -- Route suspicious queries to a specific servers pool
```

dnsdist – Examples

Dynamic blocking is handled in userspace by default, but can be done in kernel via eBPF on recent Linux kernels (4.1+)

```
function maintenance()
  addresses = exceedNXDOMAINS(100, 10) -- Get the addresses that
  ↪  generated more than 100 NXDOMAINs in the last 10 seconds
  addDynBlocks(addresses, "Exceeded NXDomain", 60) -- Block the
  ↪  addresses for a minute
end
```

dnsdist – Examples

```
nmg = newNMG()
nmg:addMask('198.51.100.0/24')
nmg:addMask('203.0.113.0/24')

selector = AndRule{QTypeRule(dnsdist.A), RegexRule('[a]{5,99}')}
 ↪  -- match QTYPE A and QNAME matching regex
selector = AndRule{selector, NetmaskGroupRule(nmg)}
 ↪  -- Add the netmask group to the rule
addAction(selector, DelayAction(100))
 ↪  -- Delay the answers to the above selector with 100 ms
```

dnsdist – Lua load balancing

```lua
function authOrRec(dq)
  if (dq.dh:getRD() == false)
  then
    return DNSAction.Pool, "auth"
  end
  return DNSAction.Pool, "recursor"
end
addLuaAction(AllRule(), authOrRec)
```

dnsdist - References

Leaseweb

Packet Clearing House

Switch

T-Mobile Czech Republic

Telepost Greenland

Transip

dnsdist

Try it!

- Packages at `https://repo.powerdns.com`
- Documentation at `http://dnsdist.org`

Thank you for your attention

Any questions?

Powered by LaTeX, based on a theme by kaszkowiak.eu