Tales of the unexpected revisited

- handling unusual DNS client behaviour (the sequel)

UKNOF31 – Cathy Almond, ISC

What is this talk about?

- Random DNS query attacks against specific domains – a quick recap
- Mitigation approaches
- Results from production environments
- Future thoughts/ideas/plans

The attack

- Attack is directed at DDOSing DNS authoritative provider, but incidentally degrades ISP resolvers in the path
- Higher query loads than usual
- Non-responding authoritative servers (directly filtering the resolvers, or simply overwhelmed)
- Increased network traffic levels

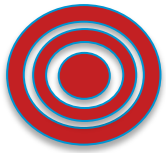
Identifying an attack

high volume of queries for non-existant sub-domains

<randomstring>.www.example.com
<anotherstring>.www.example.com

does not exist

exists



ISC

The source

- Open resolvers
  - your servers
  - your clients (CPE devices/proxies and forwarders)
- Compromised clients (botnets)
- Compromised devices

Symptoms

- Increased inbound client query traffic
- Increased outbound NXDOMAIN and SERVFAIL responses
- Resolution delays to clients
- Dropped responses
- Increased memory consumption
- Firewall connection table overflows

Evidence

- Backlog of recursive client queries
  - which queries are in the backlog?
  - is there a pattern?
  - originating from few or many clients?
- Open outbound sockets
  - to which servers; is there a pattern?
- Query logging / query-errors logging
- Network packet traces

“Do”s…

- Eliminate open resolvers
  - is yours an open resolver?
  - open client CPE devices
  - small business users forwarding local open caches to your servers
- Investigate compromised/infected clients
  - potentially several device types
  - source addresses may be spoofed

And “don’t”s…

- Panic!!
- Assume that increasing server resources (e.g. recursive client contexts, sockets, network buffers etc..) is going to help
- Block your clients (without investigating them properly first)

MITIGATION TECHNIQUES

What can we do?

What has been tried in production?

Stage 1: Lie!

- Make recursive server temporarily authoritative for the target domain
  - Local zone
  - DNS-RPZ (*qname-wait-recurse no;)

- *Manual configuration change*
- *Need to undo the mitigation afterwards*

Stage 2: Automate filtering

(Near) Real Time Block Lists

- Detect ‘bad’ domain names or just the problematic queries & filter them at ingress to the resolver
- Local auto-detection scripts
- Nominum Vantio
- BIND DNS-RPZ
- Costs associated with feeds
- Potential false-positives

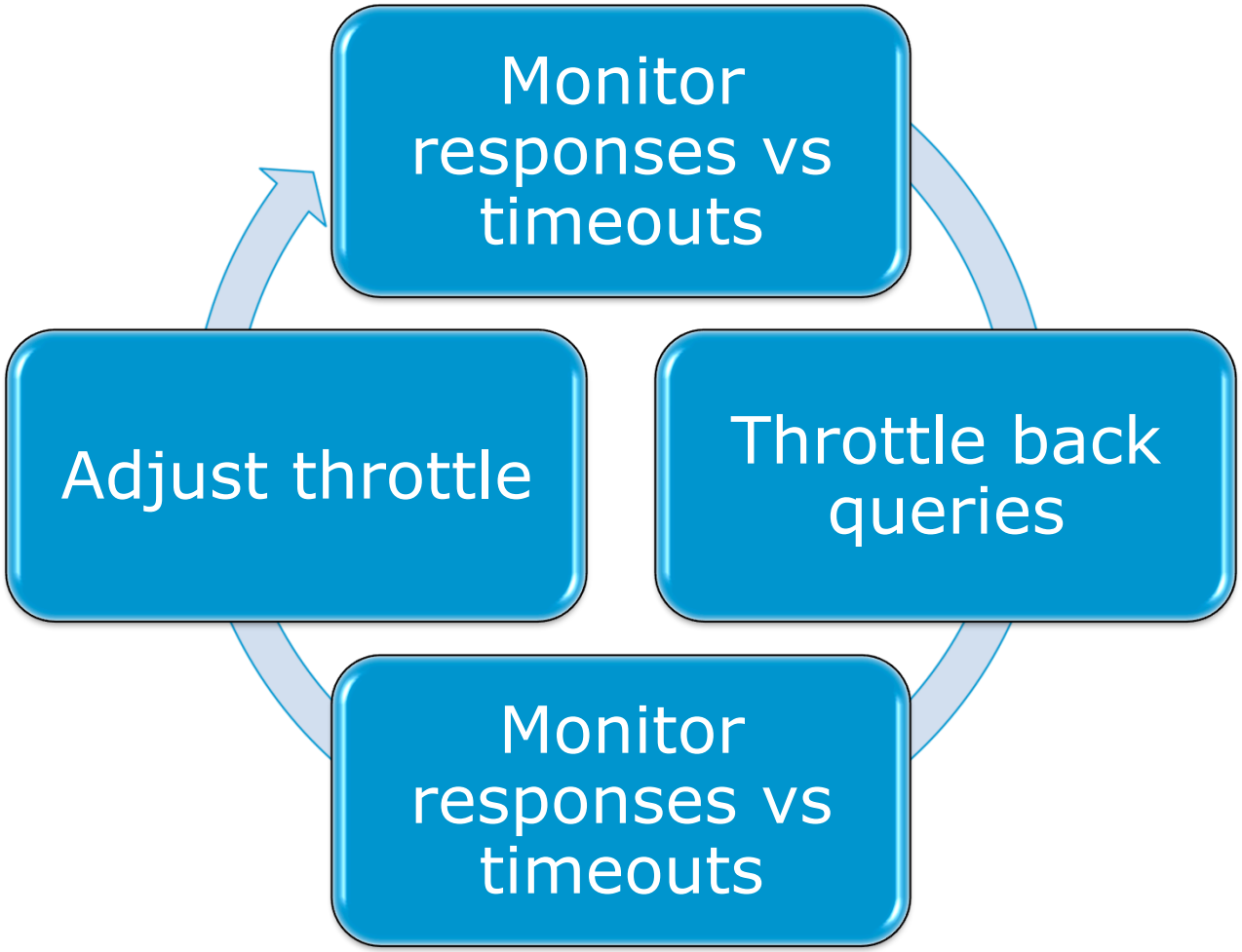
Stage 3: Tune your resolver

PER ZONE

PER SERVER

Respond SERVFAIL without waiting to timeout

Fetches-per-server



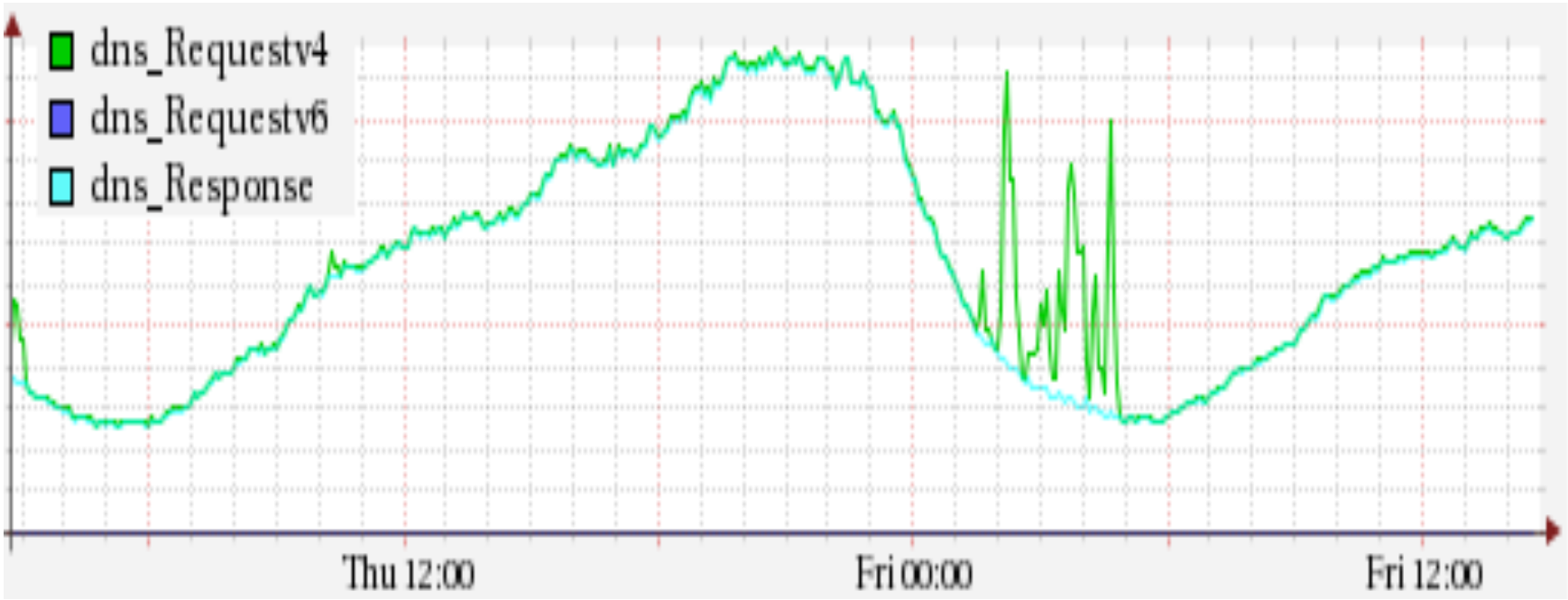
fetches-per-server

- Per-server quota dynamically re-sizes itself based on the **ratio of timeouts to successful responses**
- Completely non-responsive server eventually scales down to fetches quota of 2% of configured limit.
- Similar (loosely) in principle to what NLnet Labs is doing in Unbound

fetches-per-zone

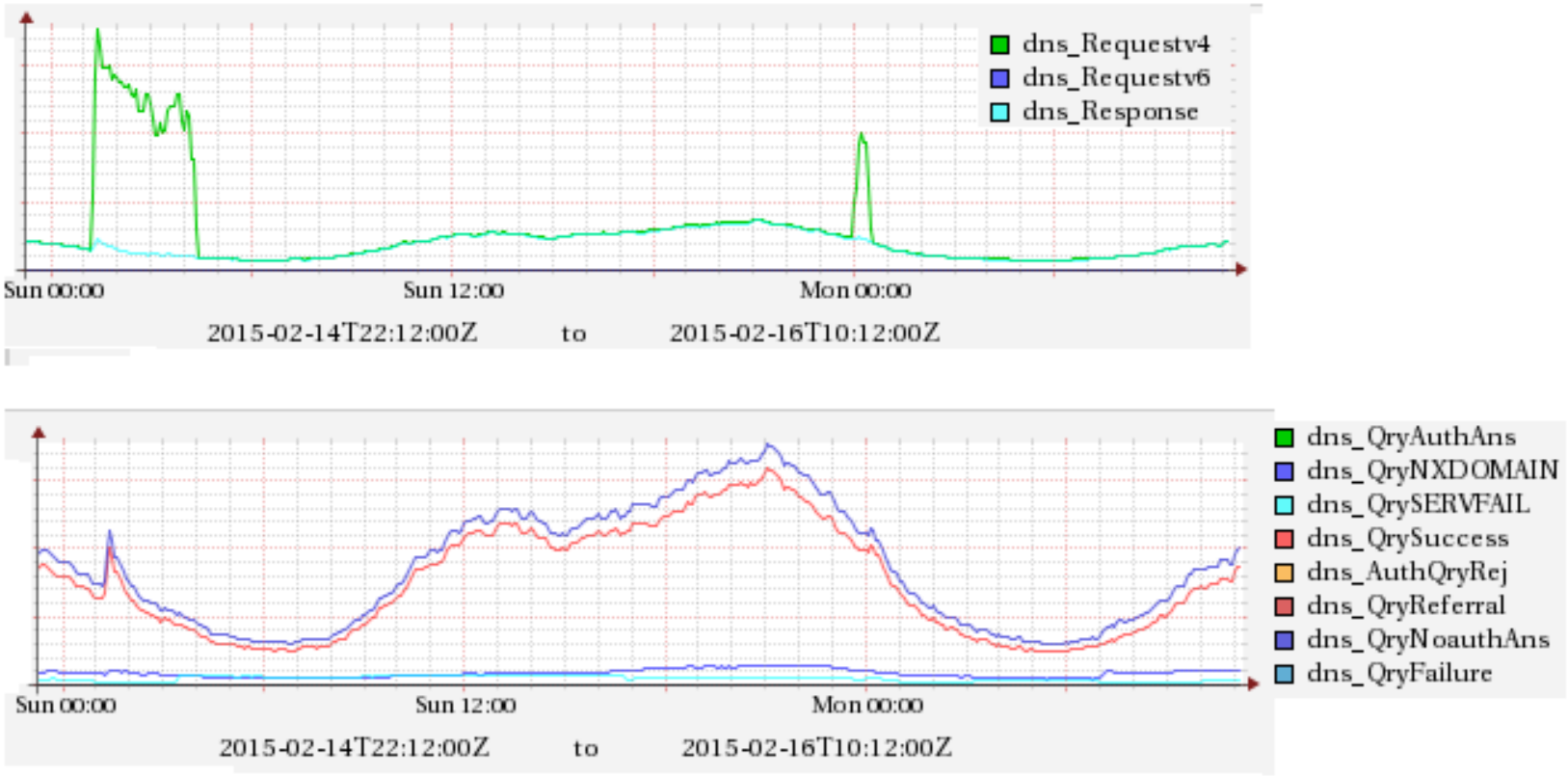
- Works with unique clients
- Default 0 (no limit enforced)
- Tune larger/smaller depending on normal QPS to avoid impact on popular domains

fetches-per-zone at Jazztel



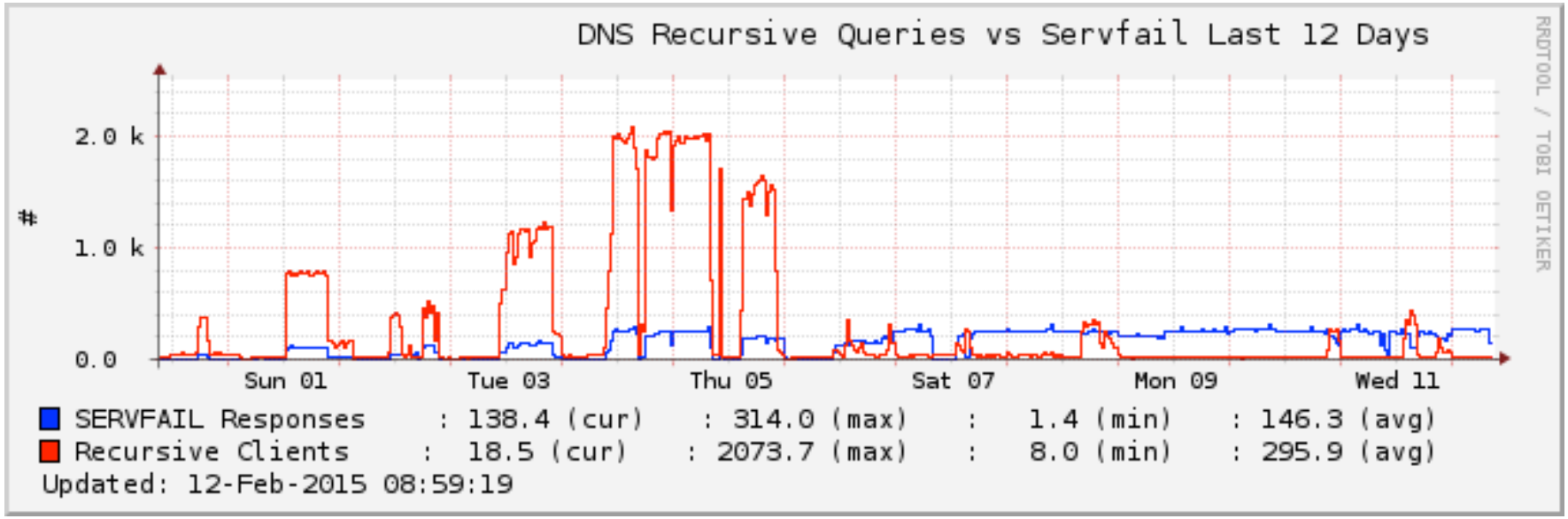
Spanish triple-play ADSL carrier & ISP
Roberto Rodriguez Navio, Jazztel Networking Engineering
used with permission

More on fetches per zone

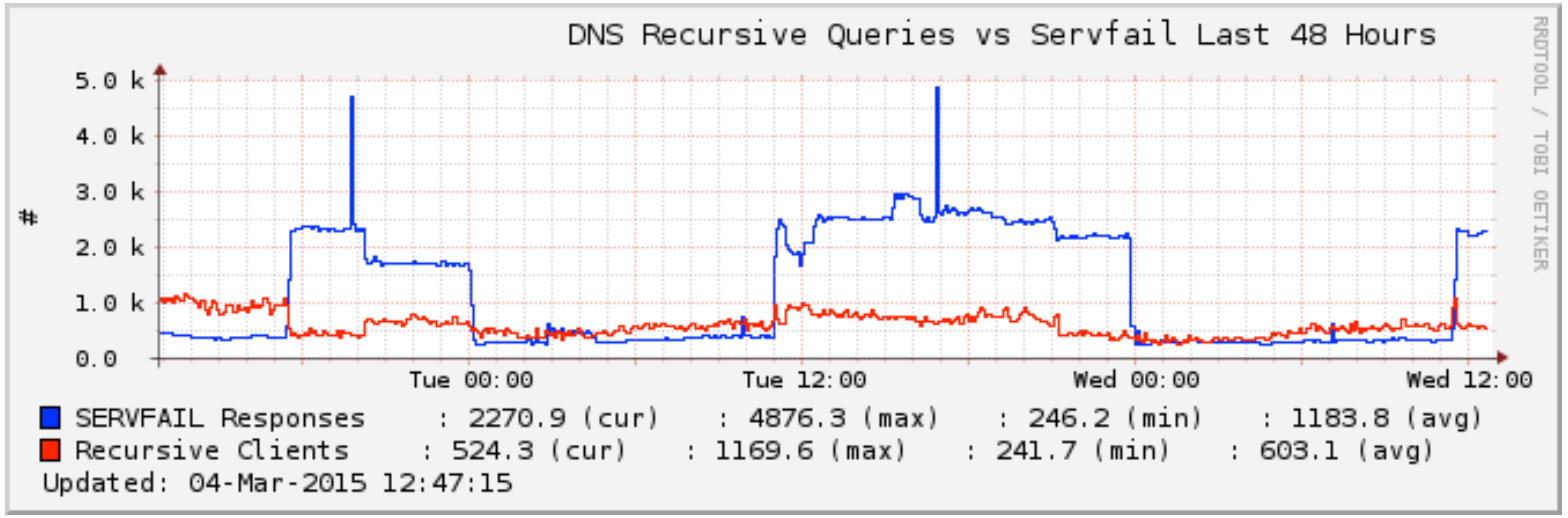


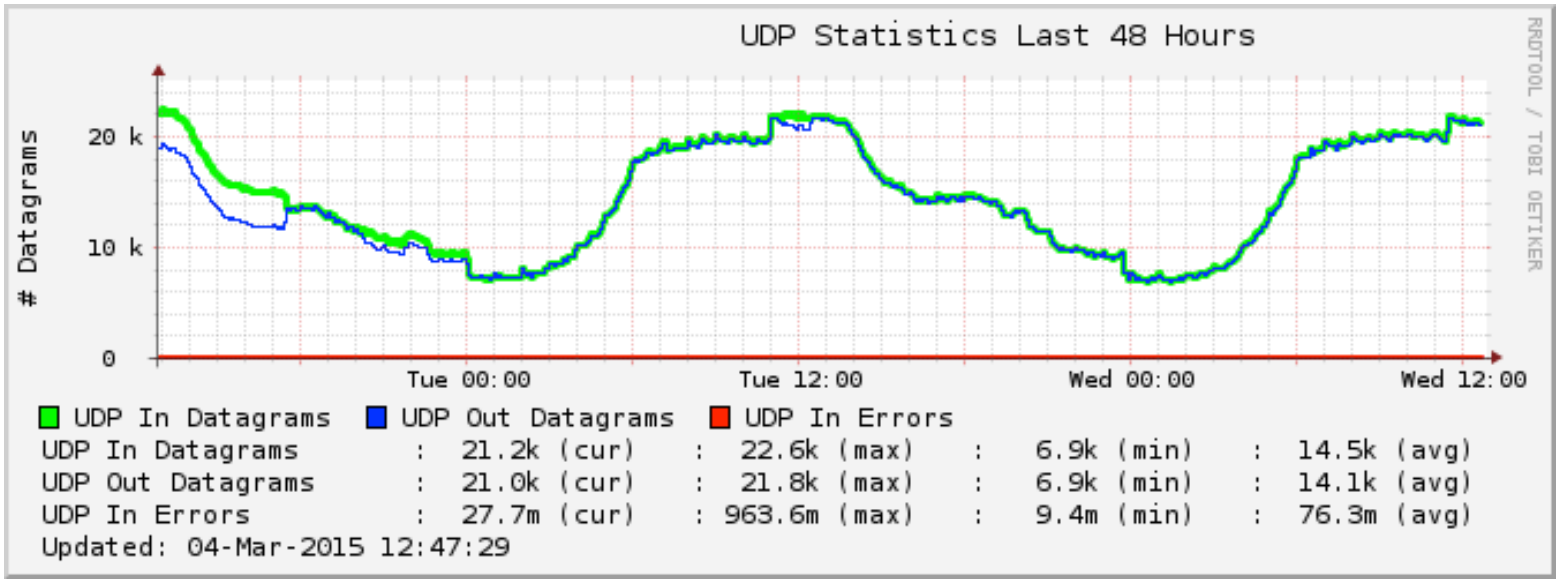
Spanish triple-play ADSL carrier & ISP
Roberto Rodriguez Navio, Jazztel Networking Engineering
used with permission

fetches-per-server



per-zone v. per-server

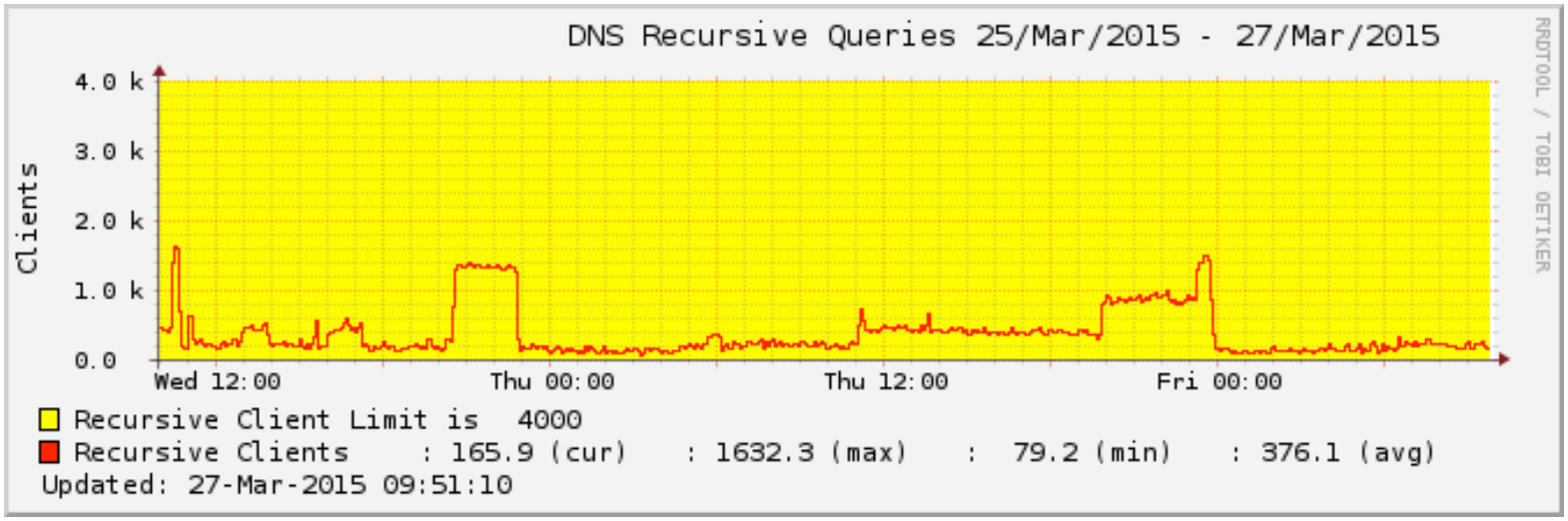


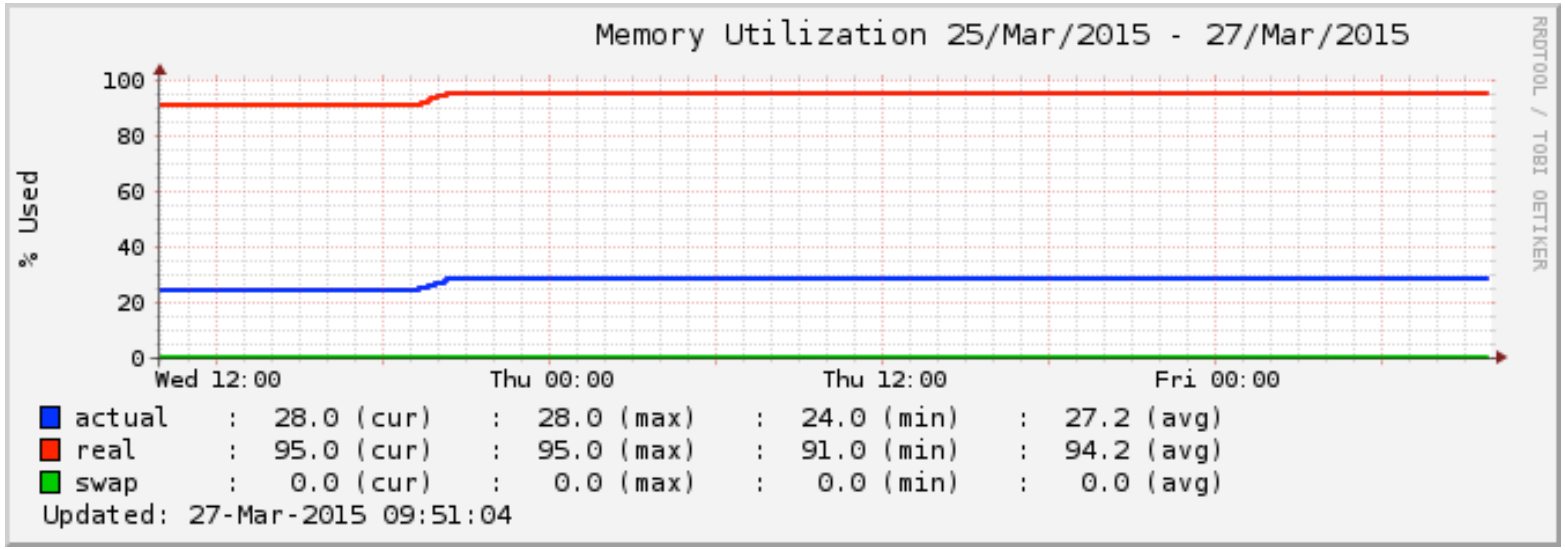


What will the user see?

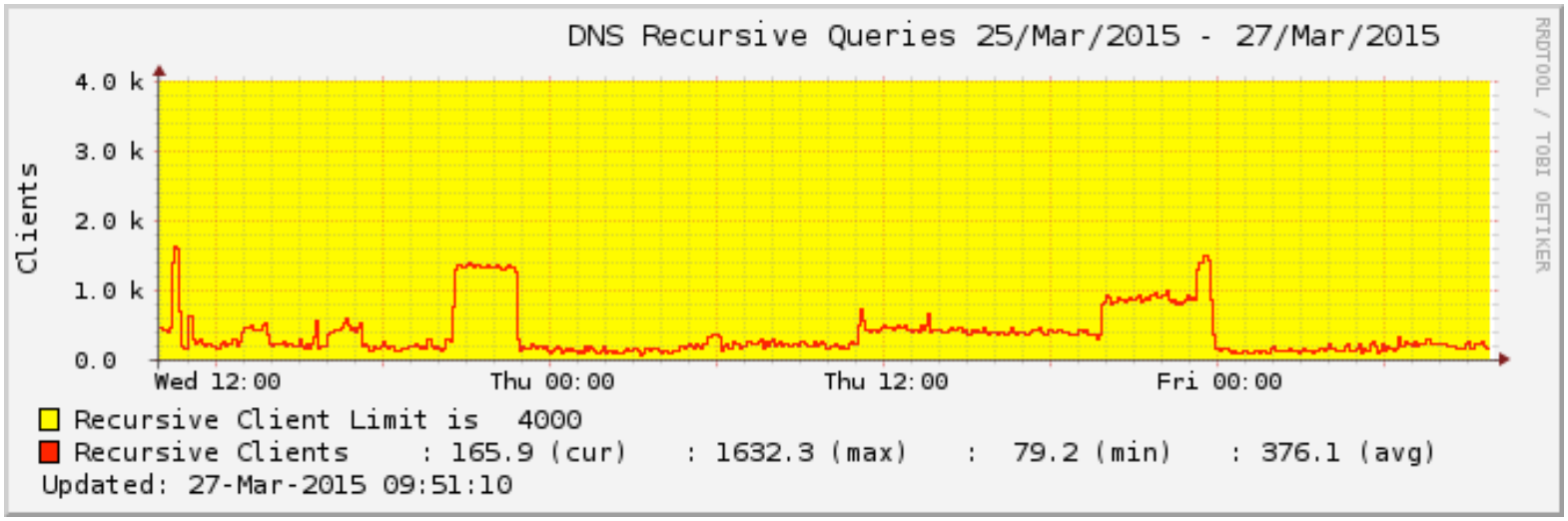
- Situation normal – no change to their usual experience (for most)
- (Some) SERVFAIL responses to names in zones that are also served by under-attack authoritative servers (collateral damage)
- NXDOMAIN responses for names in legitimate zones for which we ‘lie’

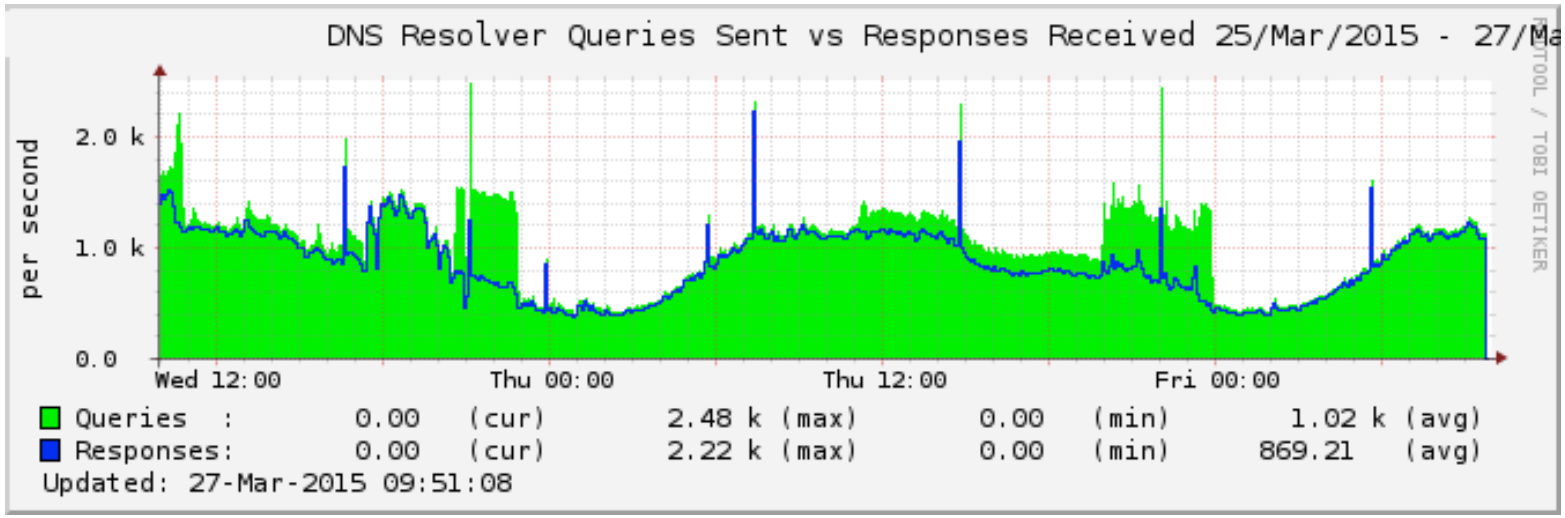
But not yet perfect…



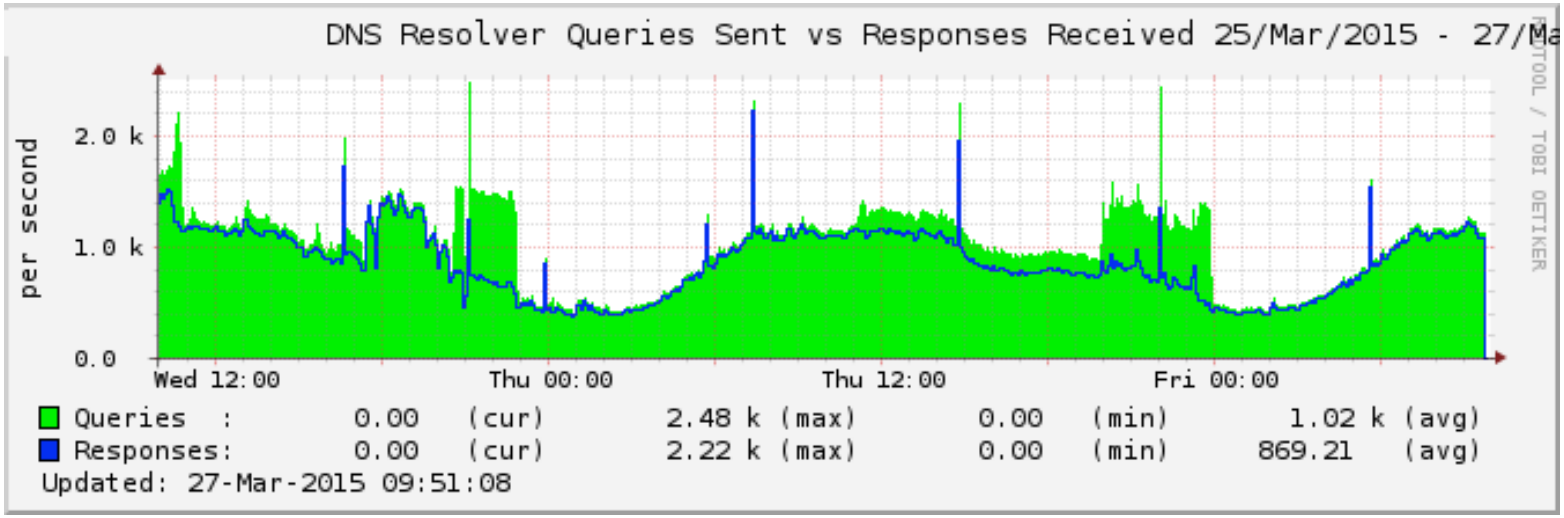


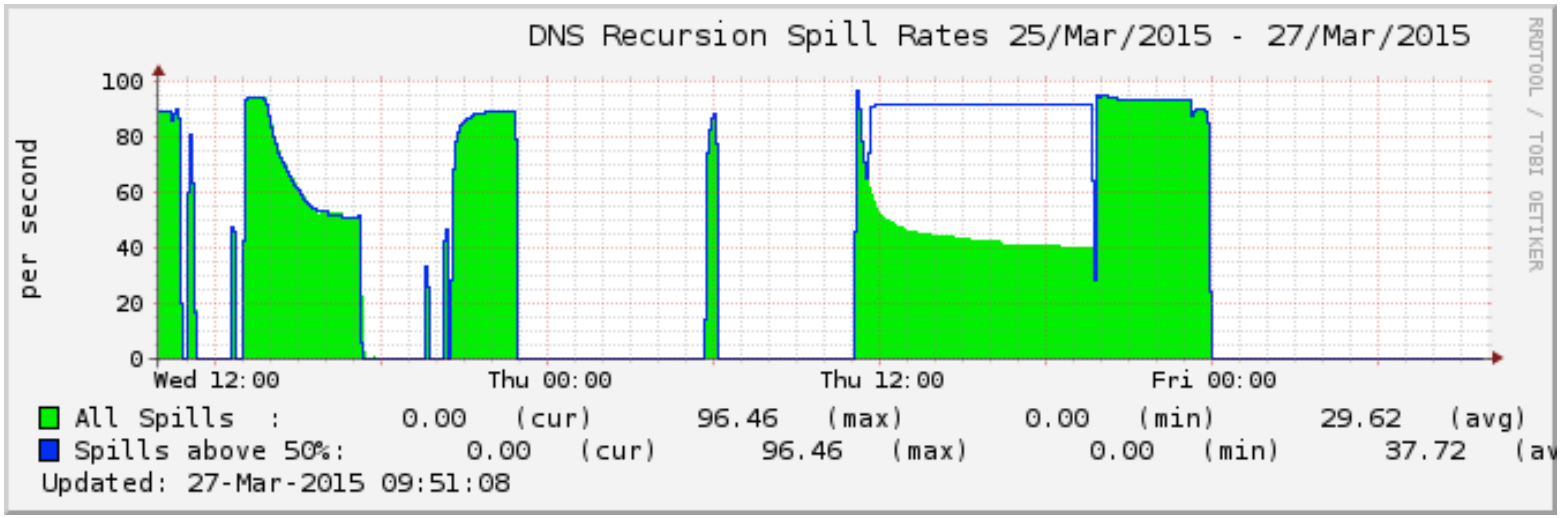
But not yet perfect…





But not yet perfect…





More ideas…

- SERVFAIL or drop (or NXDOMAIN)?
- Whitelists may be needed
- Per-server/zone override settings
- SERVFAIL cache (for client retries)
- Improved reporting & statistics
- Built-in ‘auto-DNS-RPZ’
- Persistent (non-expiring) RRsets (for ‘good’ answers)

Summary of techniques

1) Clean up your network

   eliminate open resolvers & compromised clients; look at BCP 38

2) Configure your resolver to lie

   answer authoritatively yourself; potentially automate your blacklist or subscribe to a feed for this.

3) Consider adaptive quotas

   per server; per zone

(Good feedback on these from many sources)

QUESTIONS?

bind-suggest@isc.org, cathya@isc.org