CLOUDFLARE®

Neutering ANY query: How to do it

Ólafur Guðmundsson & Filippo Valsorda

Why is ANY query bad ?

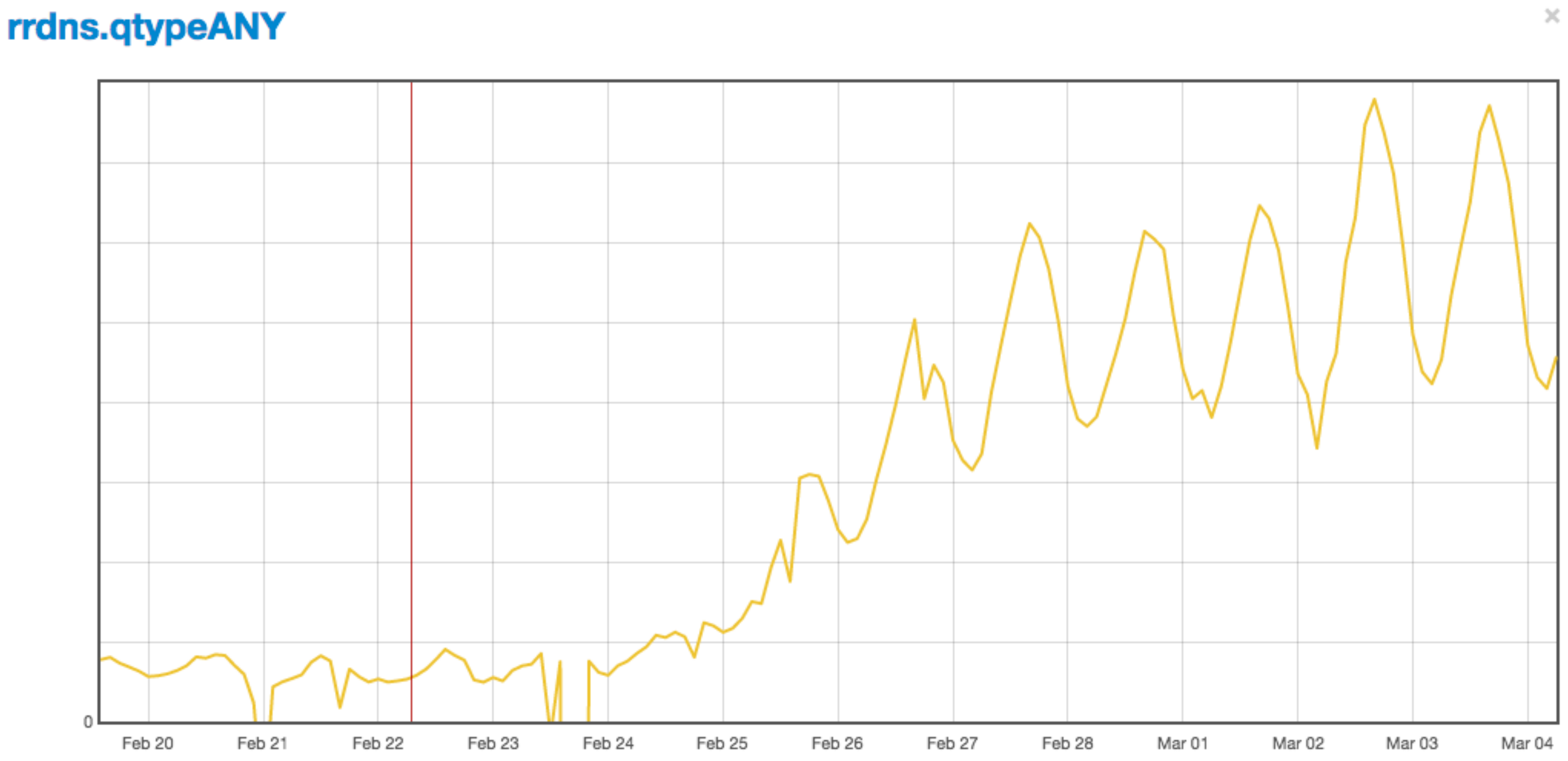
- Amplification
- Information leak
- Not reliable
- Expensive

ANY Side Effects?

- CloudFlare systems DB lookup per type
  - ==> many lookups to assemble answer
  - not all types at available at the edge

ANYone Stupid enough?

- People assume that ANY will give them ALL types at name
  - Mozilla released Firefox version that used ANY in attempt to get either A or AAAA in one query



What we did: Wrote a blog and ID

“Deprecating the DNS ANY meta-query type”

Initial Reaction: Positive

- “We have this problem”
- “We spend too much on bandwidth because of ANY queries”
- “Yes stop this information leak”

More reactions: “Hostile”

- “You are hurting Firefox and Qmail”
- “you are idiots !!!!1”
- “I use ANY to debug my systems all the time!!!”

People talked on mailing lists

- Well there should be some way to limit ANY queries
  - Great if the answer is cacheable
- ACL is debuggers friend
- Many want to do get rid of ANY queries

Why do Resolvers forward ANY query ?

Then: a word of reason

- On DNSOP mailing list DJB wrote an explanation as to what Qmail is doing:
  - https://mailarchive.ietf.org/arch/msg/dnsop/kXSApuM4i0WLoIo3_OhrCcAZ-cc
- Translation: Qmail uses ANY as a probabilistic optimization
- Will fall back to normal resolution if ANY does not yield “useful” answer

A Wise man

- At DNS-OARC in Amsterdam May 2015 Bert Hubert said

by Johnny Hart

Give me a shut-up option and I will stop asking
ADVICE
©2007 CREATORS SYNDICATE, INC. www.creators.com
3-2
hart

Copyright © 2007 Creators Syndicate, Inc.

Ways to answer ANY

- Harmful
  - RCODE != 0
    - Not Cached, resolvers tries next one and eventually fails
  - Empty answer
    - Treated as negative answer and cached for a short time
  - Referral
    - Confuses some resolvers

- Harmless
  - Existing record type
    - This works well
  - New/Unspecified Record Type
    - In almost all cases this works fine some implementations don’t like it
  - Guess what is useful
    - Some check if MX or A + AAAA are present and return only them

Did you know?

Popular Resolvers…

Seem NOT to use ANY answers to fill caches for other types

Who sends ANY questions

- Forged reflection flood
- Forged reflection flood via open resolvers w/o cache
- Forged reflection flood via open resolvers w cache
- Resolvers with empty cache when application asks
- Resolvers with empty cache when a user asks
- User sending direct query to Auth server
- Tools walking a domain or checking policies (for example spam police)
- Others?

The biggest problem: not caching resolvers

- Open Resolvers without CACHE
  - If ignored keep asking
  - If we send TC we get TCP connection
  - Majority have actual users thus **blacklisting them is not an option**
  - ***Lots of them!!!!***

Why do we care?

- Expensive and complex to enumerate all RR Type for a name
  - We hate big answers
  - Sometimes not even available => incomplete answers
- Deploying DNSSEC with on-line signing on the edge at massive scale
  - Waste of effort to sign all the RR types the query origin does not care about

Best defenses and scale

	Packets per second	Defence
Forged	Millions	Drop
Open resolvers w/o cache	Millions	Answer with something
Open resolvers w/ cache	Hundreds in bursts	Answer with something
Resolvers with empty cache	Thousands per second	Answer with something
Direct users	Tens	Human readable refusal
Tools	???	Drop

Our Way

```
$ dig example.com @ns.example.com ANY

; <<>> DiG 9.9.7-P1 <<>> example.com @ns.example.com  ANY
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36125
;; flags: qr rd; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;example.com.    IN ANY

;; ANSWER SECTION:
example.com. 3789 IN  HINFO "Please stop asking for ANY”
                            "See draft-jabley-dnsop-refuse-any"
```

Our Way: HINFO “fake” “ANY” response

- No customers use HINFO in their zones → No need for new type
- We can generate this on the fly early in the processing
  - No need for multiple **database lookups**, **discovery of all types**, or **multiple signatures**
  - Simplified our code as we can remove ANY processing from various parts
- Cached as-is by resolvers → stops retries
- Accepted by resolvers → **doesn’t break… applications**

What about DNSSEC signed zones?

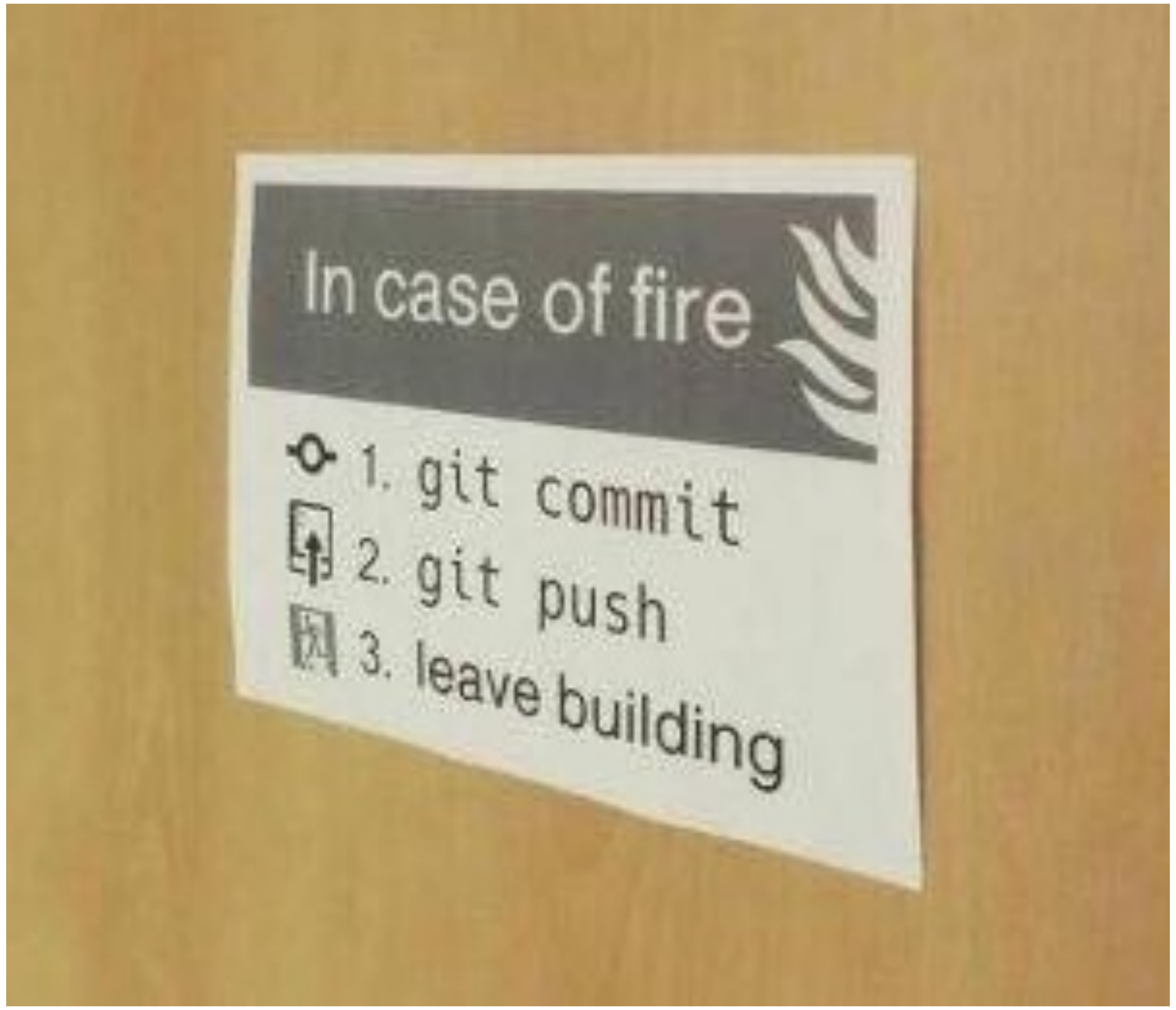
- If unsigned, validators will try all name servers and eventually go BOGUS
- BOGUS causes SERVFAIL, which breaks applications
- Need to sign the HINFO on the fly. Not hard.

What about traditional DNS servers?

- Unsigned zones: can just make up a HINFO on the fly
- Signed zones
  - Sign a HINFO for each existing name, answer normally to not existing
  - Pick one RRset and serve only that, answer normally to not existing

How to Not Break the Internet!!

- Test in a lab
- Check with reasonable people
- Measure known/possible side effects
- Test on portion of the Internet for some time
    - Back out if something… funny is seen



In case of fire

1. git commit
2. git push
3. leave building

Effects

- We have been running this for a while in a portion of our Anycast network
- No complaints (no tweets!)
- Graphs mostly unchanged

Try it

@hinfo.filippo.io

```
$ dig jgc.org ANY @hinfo.filippo.io +short
"Please stop asking for ANY" "See draft-jabley-dnsop-refuse-any"
```

- World wide proxy into that region
- Won’t be running forever
- Proxy to CloudFlare regular namesevers
- (Signed zones coming next week)

Conclusion: When hot air cools

- It is possible to reach a reasonable compromises and do simple things that address complex problems.
- By treating ANY as something we are smaller amplification reflector
- We simplified our code
- We spend less resources under attack

CLOUDFLARE®

Thanks!

Neutering ANY query: How to do it

Ólafur Guðmundsson & Filippo Valsorda