You can -j REJECT but you can not hide: Global scanning of the IPv6 Internet

Mar, 28th 2017

Tobias Fiebig
Technische Universität Berlin
Kevin Borgolte, Shuang Hao, Christopher Kruegel, Giovanni Vigna
UC Santa Barbara

Peeking at IPv6 addresses

- How can we observe IPv6 deployment?
- Several works investigate how IPv6 assignment works using large (semi/non public) datasets:
  - Foremski/Plonka/Berger: A large CDN's access logs
  - Czyz et al.: Various DNS data-sources
  - Gasser et al.: A large European IXP
- Existing techniques may miss servers and require vantage points.

Credits

- Was poked to look at this by Peter van Dijk at the last IETF in Berlin
- He is dutch, and I heard dutch people are good with DNS... so I trusted him when he suggested to look at DNS for finding IPv6 addresses.

Recap: v6 & Reverse DNS

- 128bit, 32 so called nibbles

`2001:0001:0002:0003:0004:0005:0006:0007`

- Reverse DNS
  - Map IP->FQDN
  - domain: ip6.arpa.
  - One DNS tree level per nibble:

Recap: v6 & Reverse DNS

- 128bit, 32 so called nibbles

```
2001:0001:0002:0003:0004:0005:0006:0007
7000:6000:5000:4000:3000:2000:1000:1002
```

- Reverse DNS
  - Map IP->FQDN
  - domain: ip6.arpa.
  - One DNS tree level per nibble:

Recap: v6 & Reverse DNS

- 128bit, 32 so called nibbles

- Reverse DNS
  - Map IP->FQDN
  - domain: ip6.arpa.
  - One DNS tree level per nibble:

```
2001:0001:0002:0003:0004:0005:0006:0007
7000:6000:5000:4000:3000:2000:1000:1002
7000   :6000   :5000   :4000   :3000   :2000   :1000   :1002
```

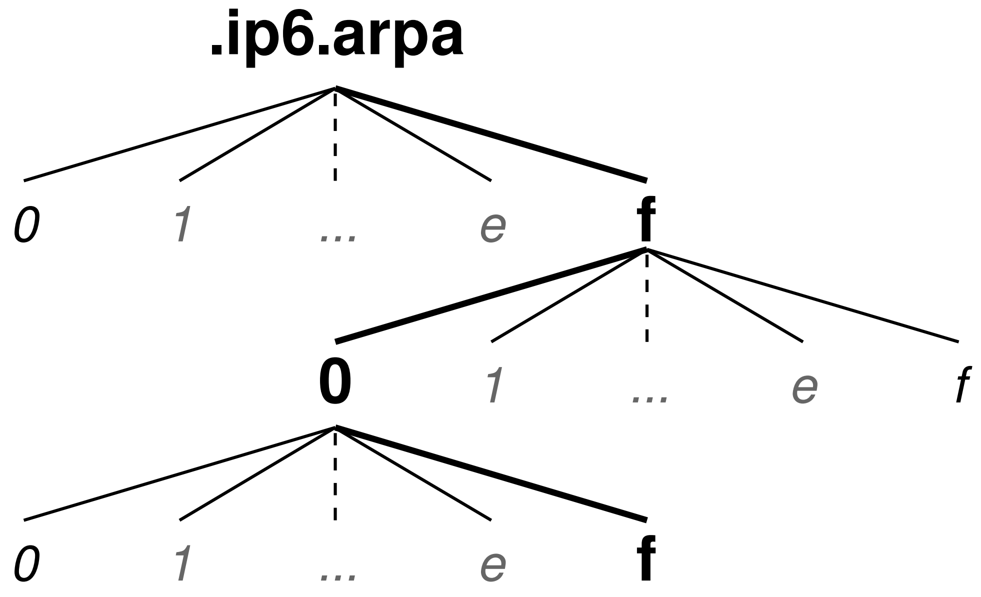
Recap: v6 & Reverse DNS

- 128bit, 32 so called nibbles
- Reverse DNS
  - Map IP->FQDN
  - domain: ip6.arpa.
  - One DNS tree level per nibble:

```
2001:0001:0002:0003:0004:0005:0006:0007
7000:6000:5000:4000:3000:2000:1000:1002
7000    :6000    :5000    :4000    :3000    :2000    :1000    :1002
7.0.0.0.6.0.0.0.5.0.0.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.1.0.0.2.ip6.arpa.
```

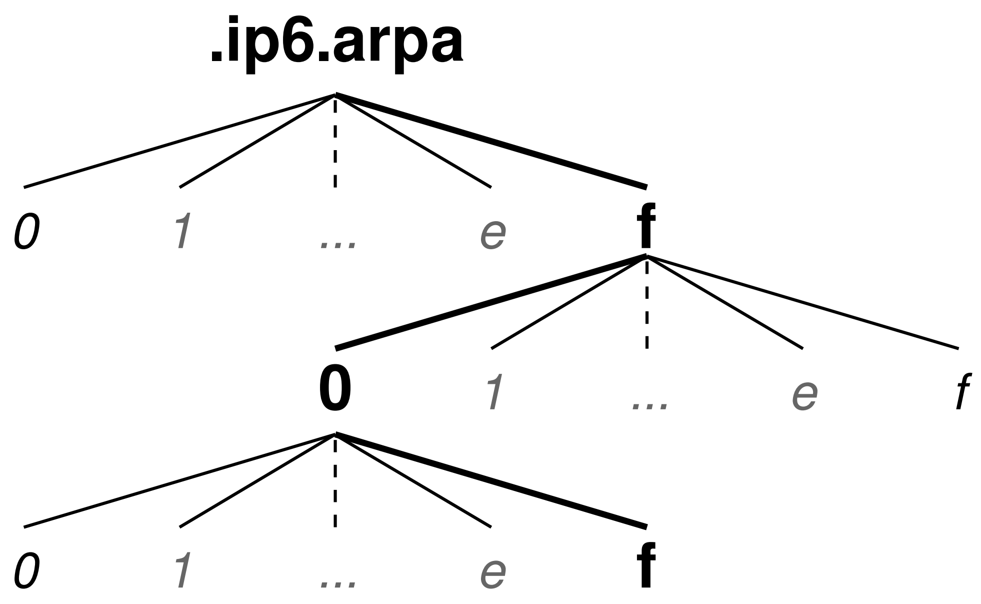
Basic Technology

- Technically (RFC1034, clarified in RFC8020) NXDOMAIN means “there is nothing here or anywhere thereunder in the tree”.
- So we only enter branches, if we get a NOERROR



Basic Technology

- In itself not new:
  - RFC7707 (IPv6 discovery techniques) explicitly mentions this technique and Peter van Dijk's blog article
  - Van Dijk himself found some even earlier works: http://7bits.nl/blog/posts/ip6-arpa-prior-art-and-results
  - There is even a brief python implementation by him, which I used as a starting ground: https://github.com/habbie/ip6-arpa-scan/



Breadth (a little) first

- Lets split up the zones:

  Example address:

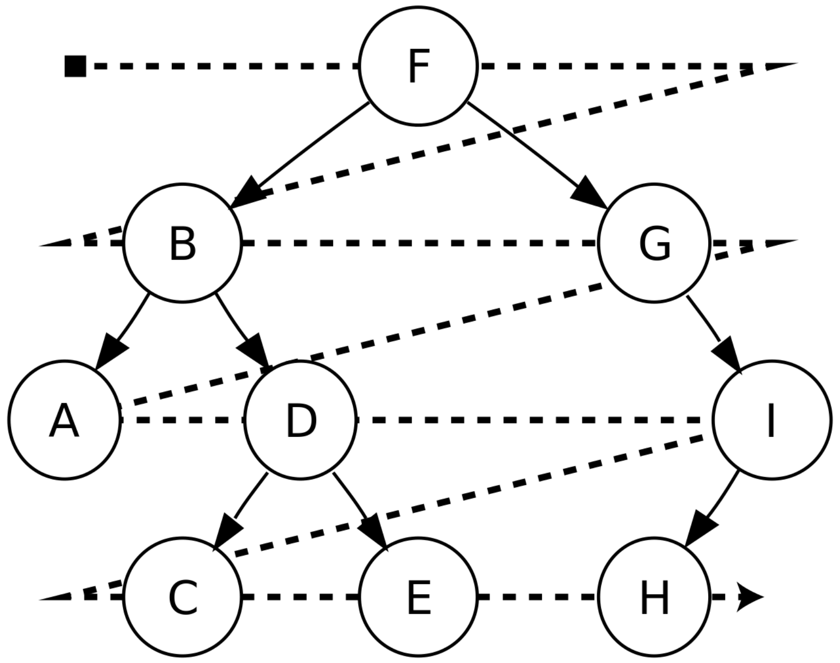
  2001:0001:0002:0003:0004:0005:0006:0007 < use 16 bit/4 nibble delimiters

  - -> 7.0.0.0.6.0.0.0.5.0.0.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.1.0.0.2.ip6.arpa.

  - Start at ip6.arpa.
  - Enumerate to a depth of 16bits (4 nibbles)
  - for each results (e.g. 1.0.0.2.ip6.arpa., 4.a.0.2.ip6.arpa.):

    Enumerate to a depth of 16bits more
  - Rinse and repeat until you hit 128bit

    (optionally, go directly from 64bit to 128bit)

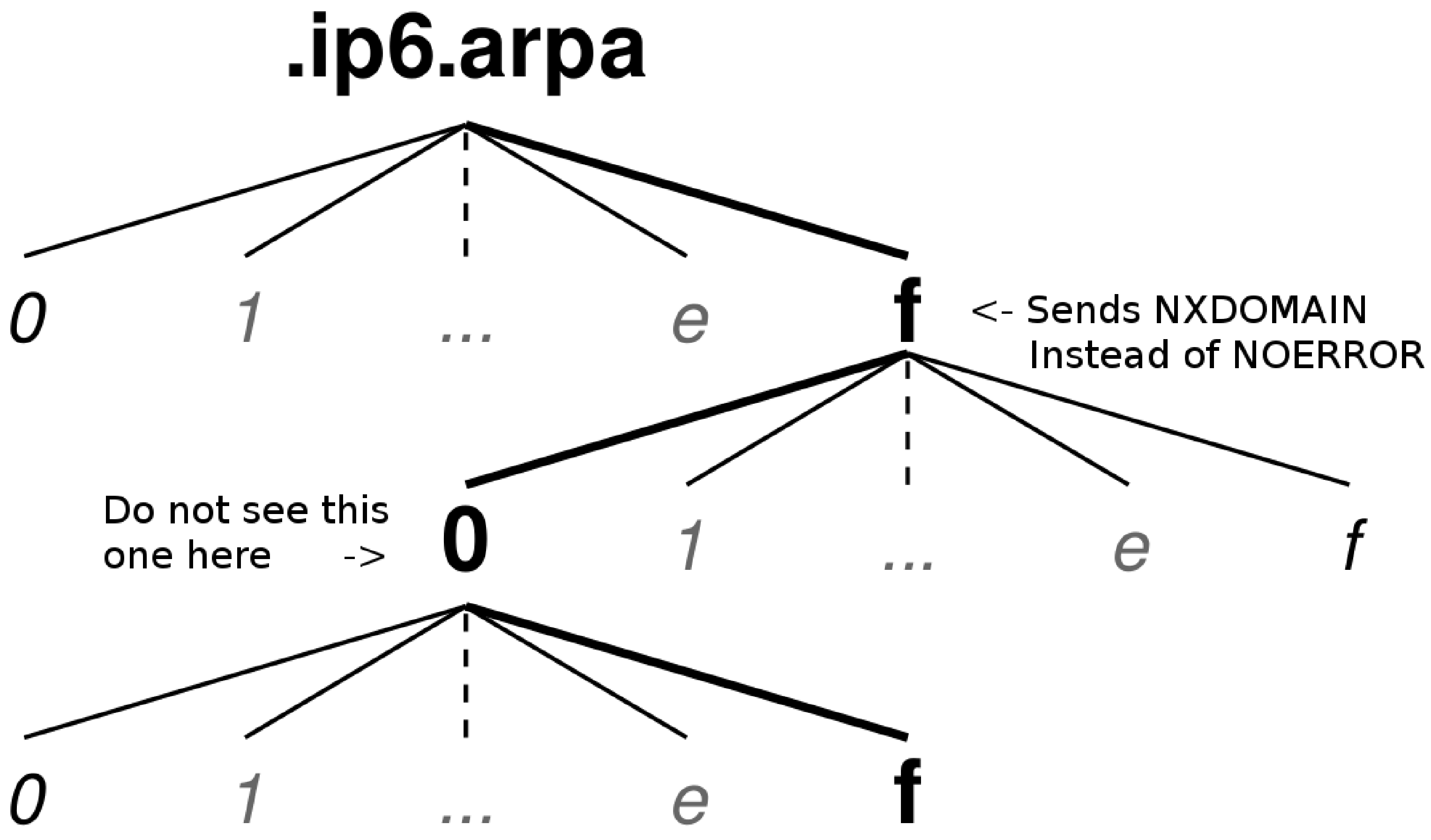


Detecting Auto-Gen Zones

- Before each step (e.g. with the previously enumerated /64's):
  - Check for i in {0..f} if
    ```
    f.f.f.f.f.f.f.f.f.f.f.f.f.f.f.f.3.0.0.0.2.0.0.0.1.0.0.0.1.0.0.2.ip6.arpa.
    e.e.e.e.e.e.e.e.e.e.e.e.e.e.e.e.3.0.0.0.2.0.0.0.1.0.0.0.1.0.0.2.ip6.arpa.
    ...
    0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.1.0.0.2.ip6.arpa.
    ```
  - exit
- If at least three (personal preference) exist, the zone is not enumerated
- Ideally go for four-nibble steps, and do an auto-gen check at every step!

Non RFC8020 Compliance

- Servers that sent NXDOMAIN instead of NOERROR for non-existing nodes with children:



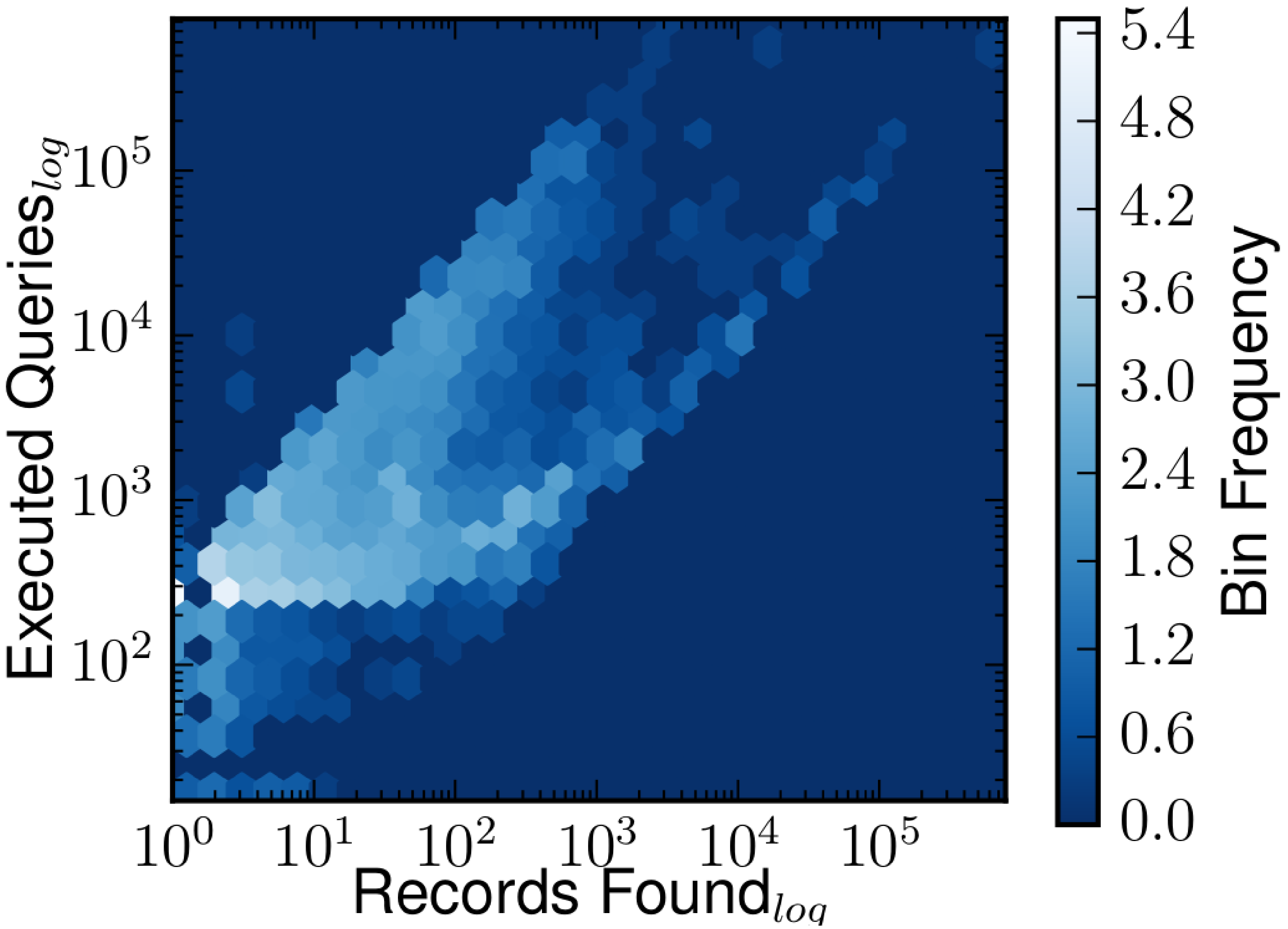
Non RFC8020 Compliance

- Solution: Seeding, using an aggregated BGP view
- Algorithm:
  - In: 2001:1:2:3::/64 as seed from BGP
  - Out:

```
3.0.0.0.2.0.0.0.1.0.0.0.1.0.0.2.ip6.arpa.
        2.0.0.0.1.0.0.0.1.0.0.2.ip6.arpa.
                1.0.0.0.1.0.0.2.ip6.arpa.
                        1.0.0.2.ip6.arpa.
```

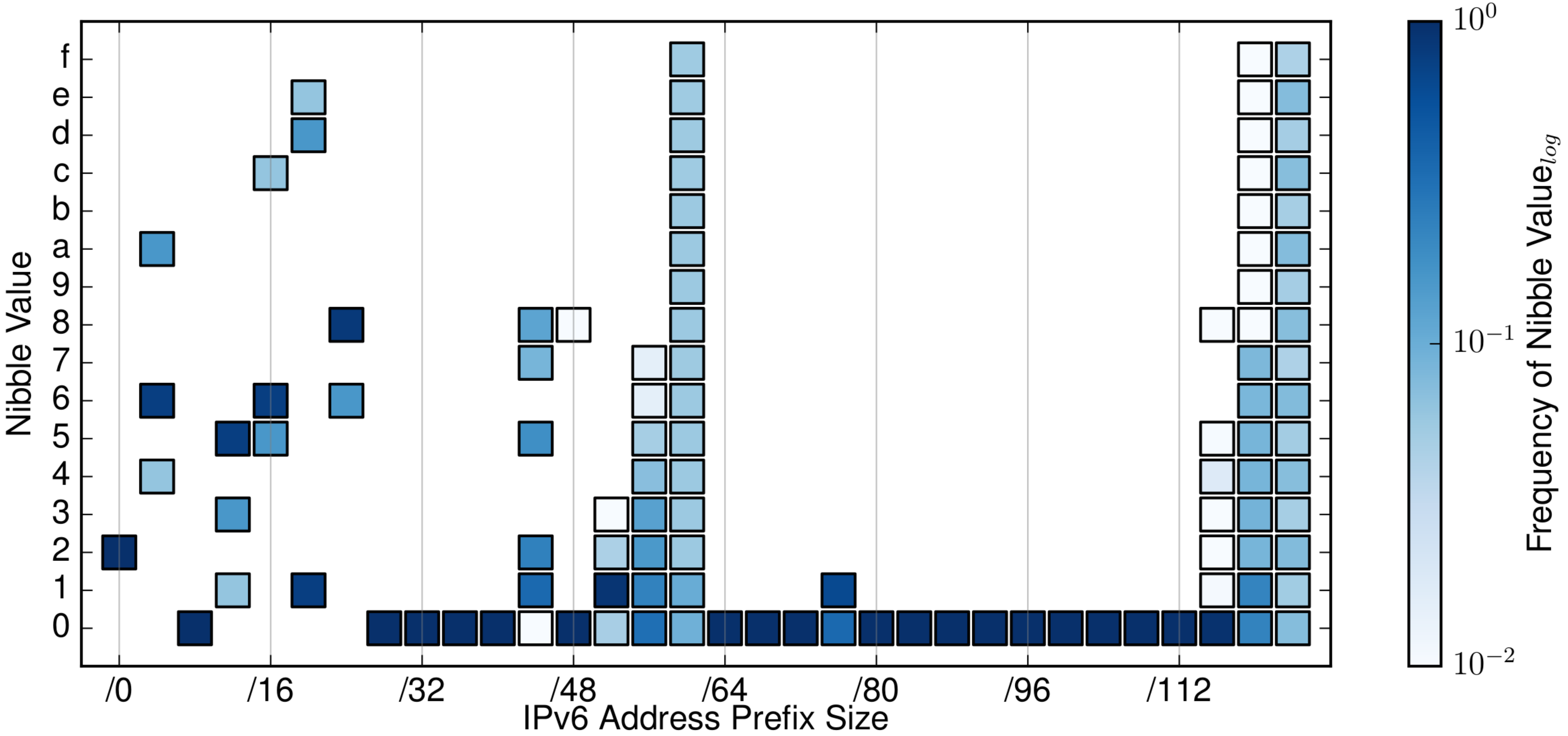
- Other possible sources:
  - Czyz et al.'s PTRv4 -> AAAA lookups
  - All v6 Datasets you can get your hands on!

First insights...

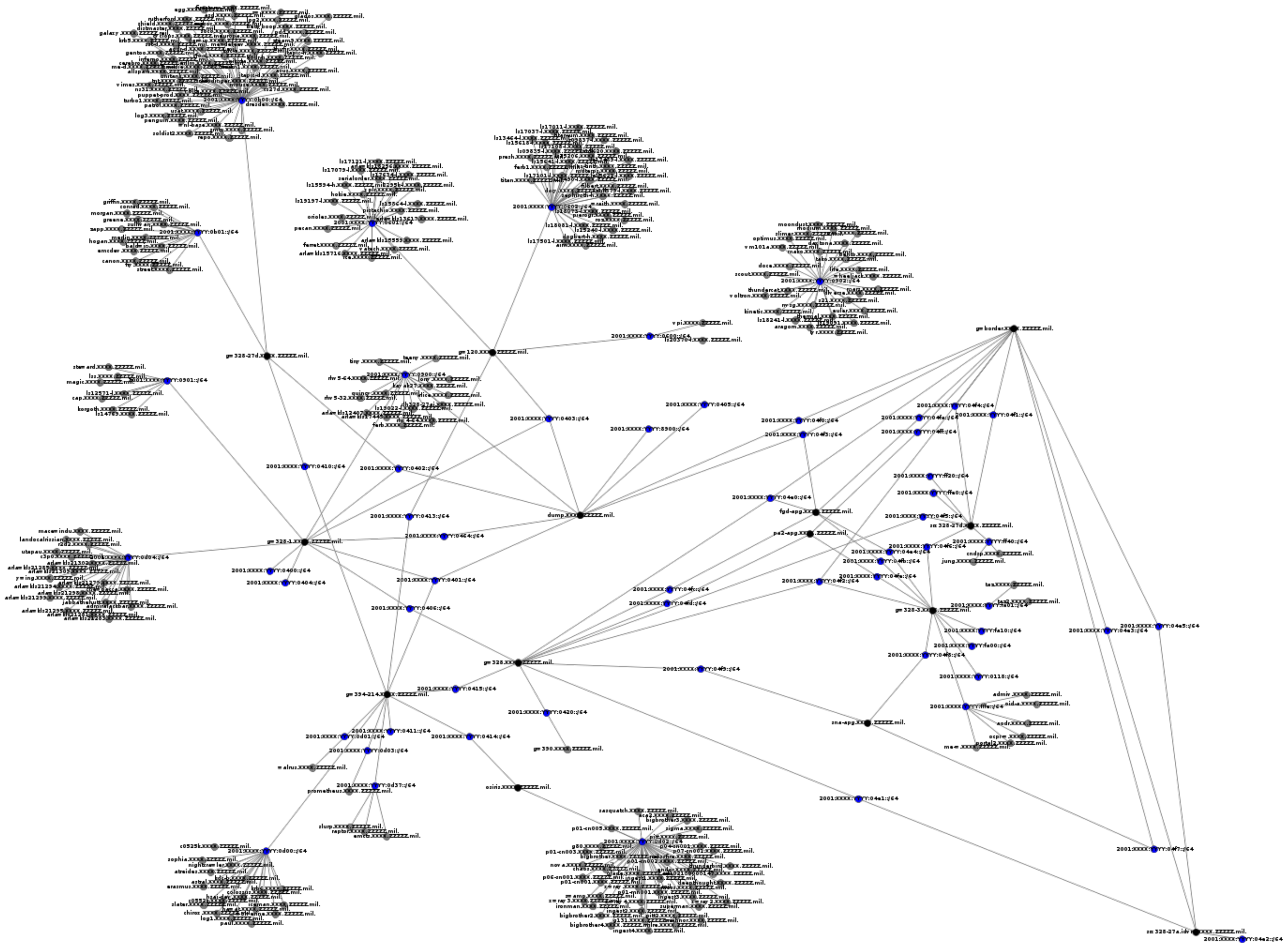


/64 -> /128

My little SaaS provider (i)



Zoomed topology



Mitigation

- Simon Arlott suggests using wildcard RRs
  - He wrote a tool: https://github.com/lp0/ip6walk
- By then (2012) only applicable to ldns (1.6.12) NSD (3.2.10), according to van Dijk
- But the concept should be clear...

Conclusion

- You can -j REJECT but you can not hide ;-)
- Toolchain: https://gitlab.inet.tu-berlin.de/ptr6scan/toolchain

  (beware: Academic code... like startup code, but we do not call it production ready...)
- Publication:

  "Something From Nothing (There): Collecting Global IPv6 Datasets From DNS", *T. Fiebig, K. Borgolte, S. Hao, C. Kruegel, G. Vigna*, accepted for PAM, Sydney, Australia, March 30-31, 2017
- Full-length talk: https://media.ccc.de/v/33c3-8061-you_can_-j_reject_but_you_can_not_hide_global_scanning_of_the_ipv6_internet