A Look at RFC 8145 Trust Anchor Signaling for the 2017 KSK Rollover

Duane Wessels

October 11, 2017

Background

2017 Root Zone KSK Rollover

- ~~October 11, 2017!~~
- Root zone DNSKEY RRset signatures generated from KSK-2017.
- Validating name servers require updated trust anchors before then.
- It would be really nice to know if validators update their trust anchors.

What Is A Trust Anchor?

RFC 4033:

> "A configured DNSKEY RR or DS RR hash of a DNSKEY RR. A validating security-aware resolver uses this public key or hash as a starting point for building the authentication chain to a signed DNS response. In general, a validating resolver will have to obtain the initial values of its trust anchors via some secure or trusted means outside the DNS protocol. Presence of a trust anchor also implies that the resolver should expect the zone to which the trust anchor points to be signed."

How Are Trust Anchors Updated?

- RFC 5011 “Automated Updates of DNS Security (DNSSEC) Trust Anchors.”
- Operating System updates.
- Manually by a system administrator.

How Can We Tell If Trust Anchors Are Updated?

- Can we query all validators, and ask for their trust anchor?
  - Not really.
  - Only Unbound supports a DNS query to observe its trust anchor:
    - **trustanchor.unbound CH TXT** as of v1.6.2
  - They should have ACLs to block external queries anyway.
- How about a “sentinel” record signed by only the new KSK?
  - If the old KSK signs the new KSK (which it must), then new KSK is trusted for validation even if it’s not in the trust anchor set.
  - Also complicated due to root zone DNSSEC design.
- Have validators self-report?

RFC 8145 -- Signaling Trust Anchor Knowledge in DNS Security Extensions (DNSSEC)

RFC 8145 – Key Tag Signaling

- Validators periodically report trust anchor key tags.
- What's a key tag?
  - A 16-bit integer that identifies and enables efficient selection of DNSSEC public keys. Much like a ones' complement checksum.
  - 19036 – key tag for KSK-2010
  - 20326 – key tag for KSK-2017
- Reported to a zone's authoritative name servers.
- Should be transmitted about as frequently as DNSKEY expire.

Two Forms of Key Tag Signaling

- edns-key-tag option.
  - An appended option code in the ENDS0 / OPT record
- Separate key tag query.
- Key tag encoded in query name, using hexadecimal representation.
  - 19036 = hex 4a5c
  - 20326 = hex 4f66

Timeline & Implementations

When	What
2015 December	draft-ietf-dnsop-edns-key-tag-00
2016 July	First implementation in BIND
2017 February	draft-ietf-dnsop-edns-key-tag-05
2017 April	RFC 8145
2017 April	First implementation in Unbound
2017 May	Start collecting data

BIND: ‘trust-anchor-telemetry’ defaults to ‘yes’

Unbound: initially ‘trust-anchor-signaling’ defaults to ‘no’, changed to ‘yes’ around October 1, 2017

EDNS0 vs Qname Key Tag Signals

- BIND and Unbound implement qname-based signaling.
- Any evidence of the edns-key-tag option code (14)?
- Scanned 7 days of pcap files
- Found TWO packets with EDNS0 option edns-key-tag!
  - But really looks like COOKIE (10); optionlen = 8, versus 2
  - Bad UDP checksum
  - → bitflip in option code

- Qname wins!

Data

Data Sources

- Key Tag signals are sent to the name servers authoritative for the key they represent.
- In this case, the root zone.
- This data comes from A-root and J-root.
- Selection bias caveat: data provided by only relatively recent implementations.

Data Sample

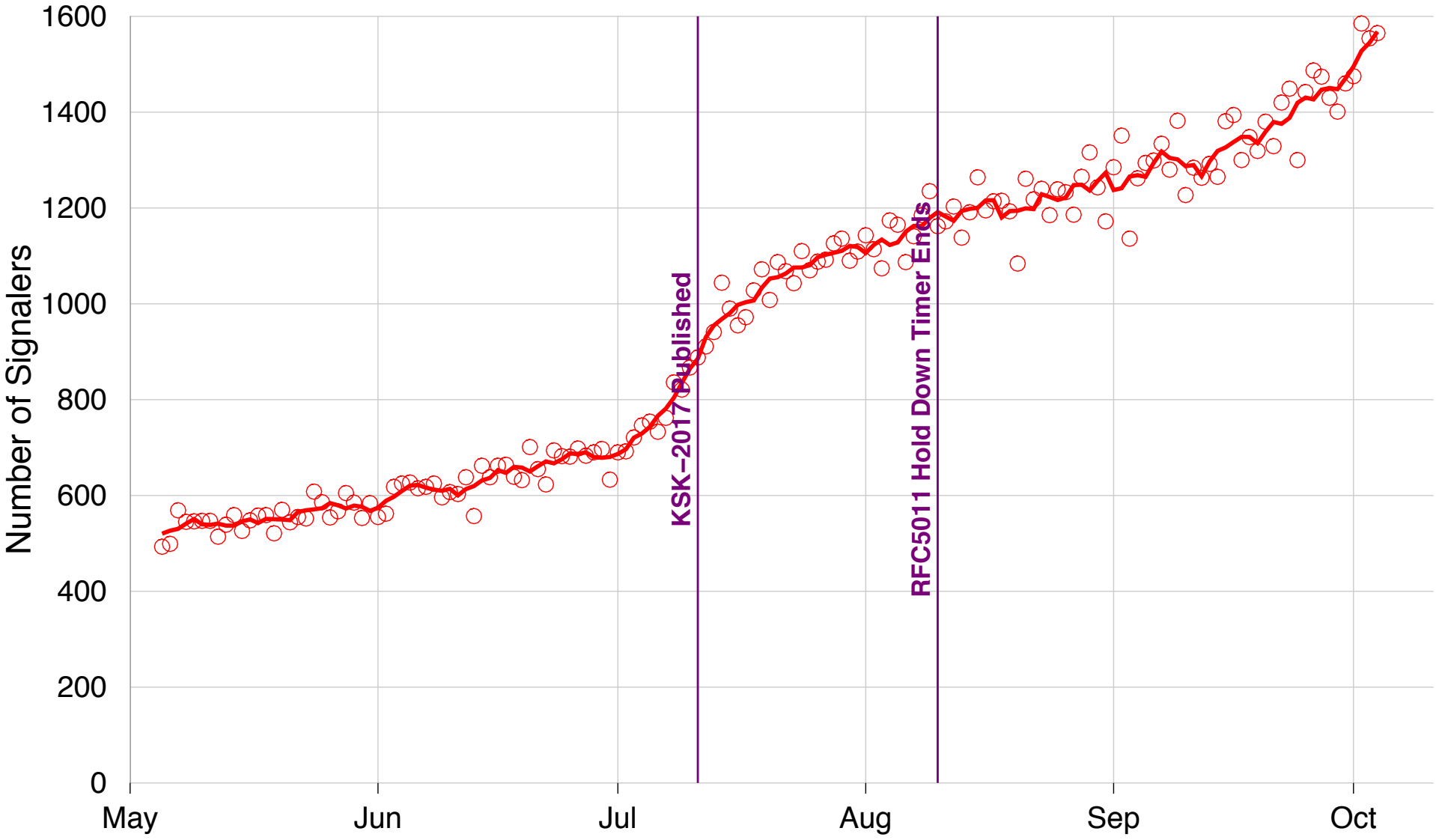
```
SELECT `timestamp`,lower(qname),dstip,srcip,year,month,day
FROM some_hadoop_hive_table
WHERE lower(qname) rlike '^_ta-'
AND qtype = 10
AND product = 'root';
```

```
1500479443  _ta-4a5c       128.x.x.x    192.58.128.30        2017 7 19
1500439539  _ta-4a5c       2a00:x:x::x  2001:503:ba3e::2:30  2017 7 19
1500476401  _ta-4a5c       2001:x:x::x  2001:503:c27::2:30   2017 7 19
1500476401  _ta-4a5c       2001:x:x::x  2001:503:c27::2:30   2017 7 19
1500495841  _ta-4a5c-4f66  188.x.x.x    198.41.0.4           2017 7 19
1500464521  _ta-4a5c       5.x.x.x      192.58.128.30        2017 7 19
1500476401  _ta-4a5c       2001:x:x::x  2001:503:c27::2:30   2017 7 19
1500476401  _ta-4a5c       194.x.x.x    198.41.0.4           2017 7 19
1500476401  _ta-4a5c       2001:x:x::x  2001:503:c27::2:30   2017 7 19
1500476401  _ta-4a5c       194.x.x.x    198.41.0.4           2017 7 19
1500495841  _ta-4a5c-4f66  188.x.x.x    198.41.0.4           2017 7 19
```

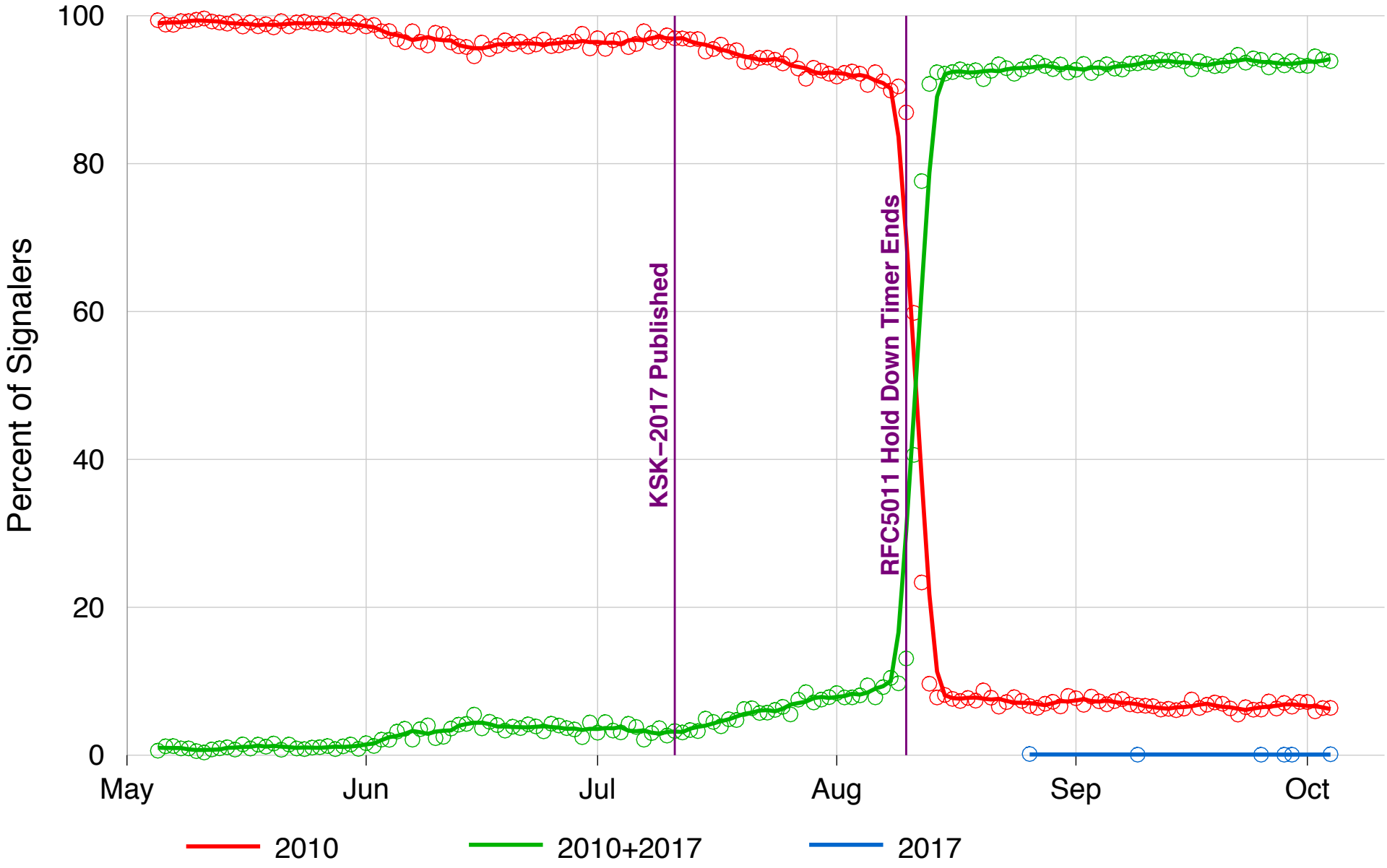
Data Processing

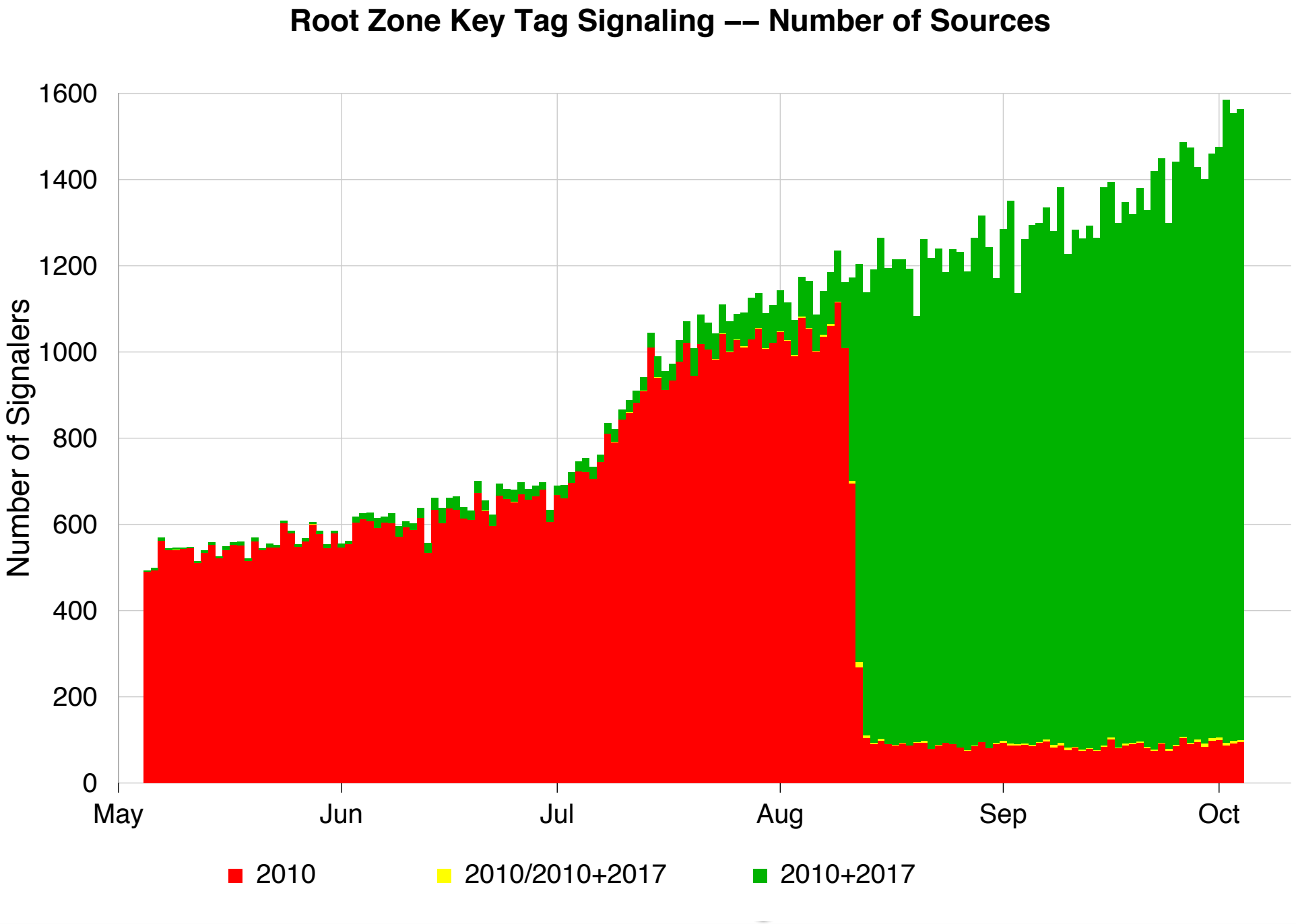
- For each day...
- Find key tag queries...
- For only the root zone...
- Count number of source IPs whose key tags contain:
  - KSK-2010 only
  - KSK-2017 only
  - KSK-2010 AND KSK-2017
  - KSK-2010 OR KSK-2017

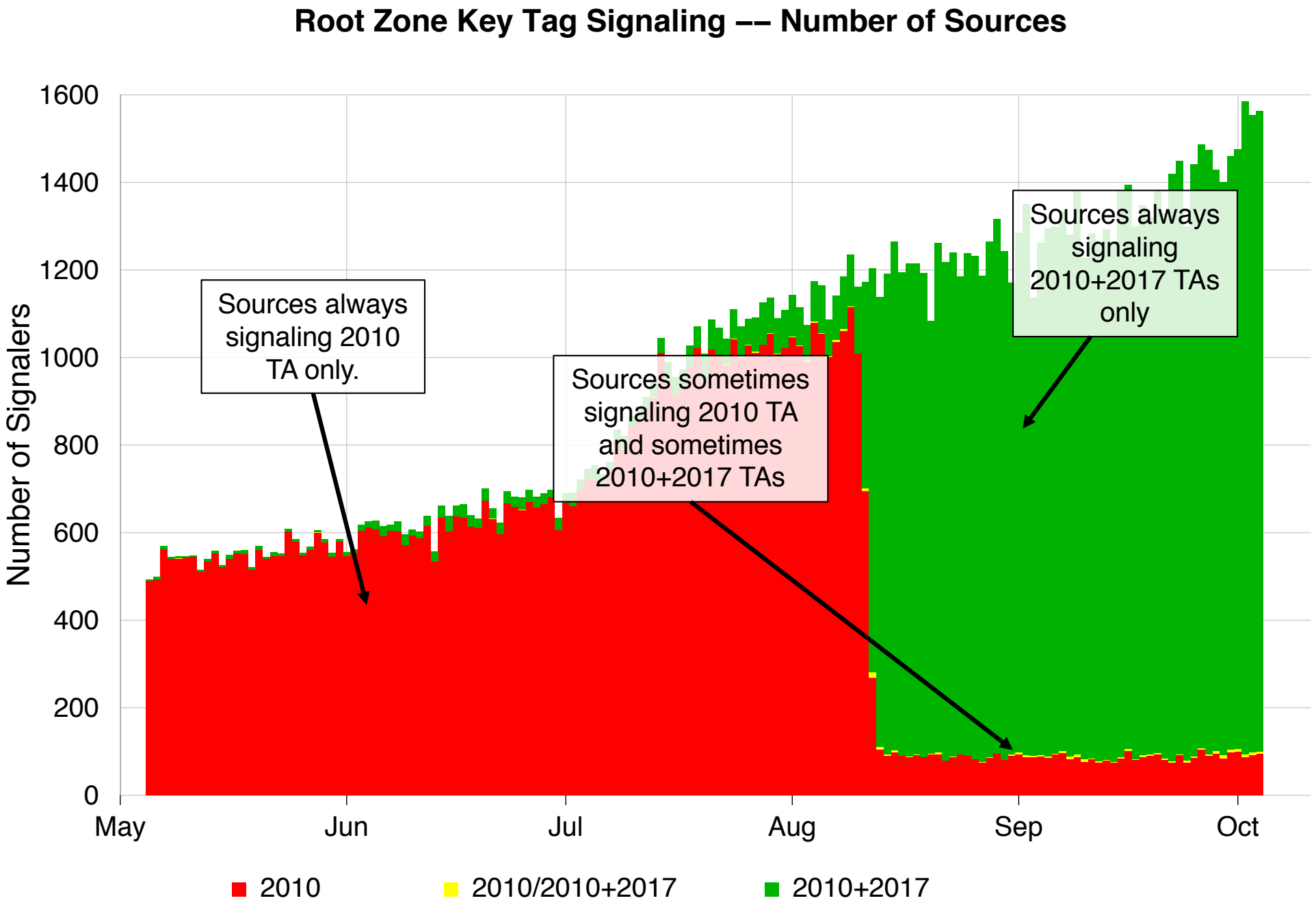
Root Zone Key Tag Signaling –– Number of Sources



Root Zone Key Tag Signaling –– TA Update Evidence



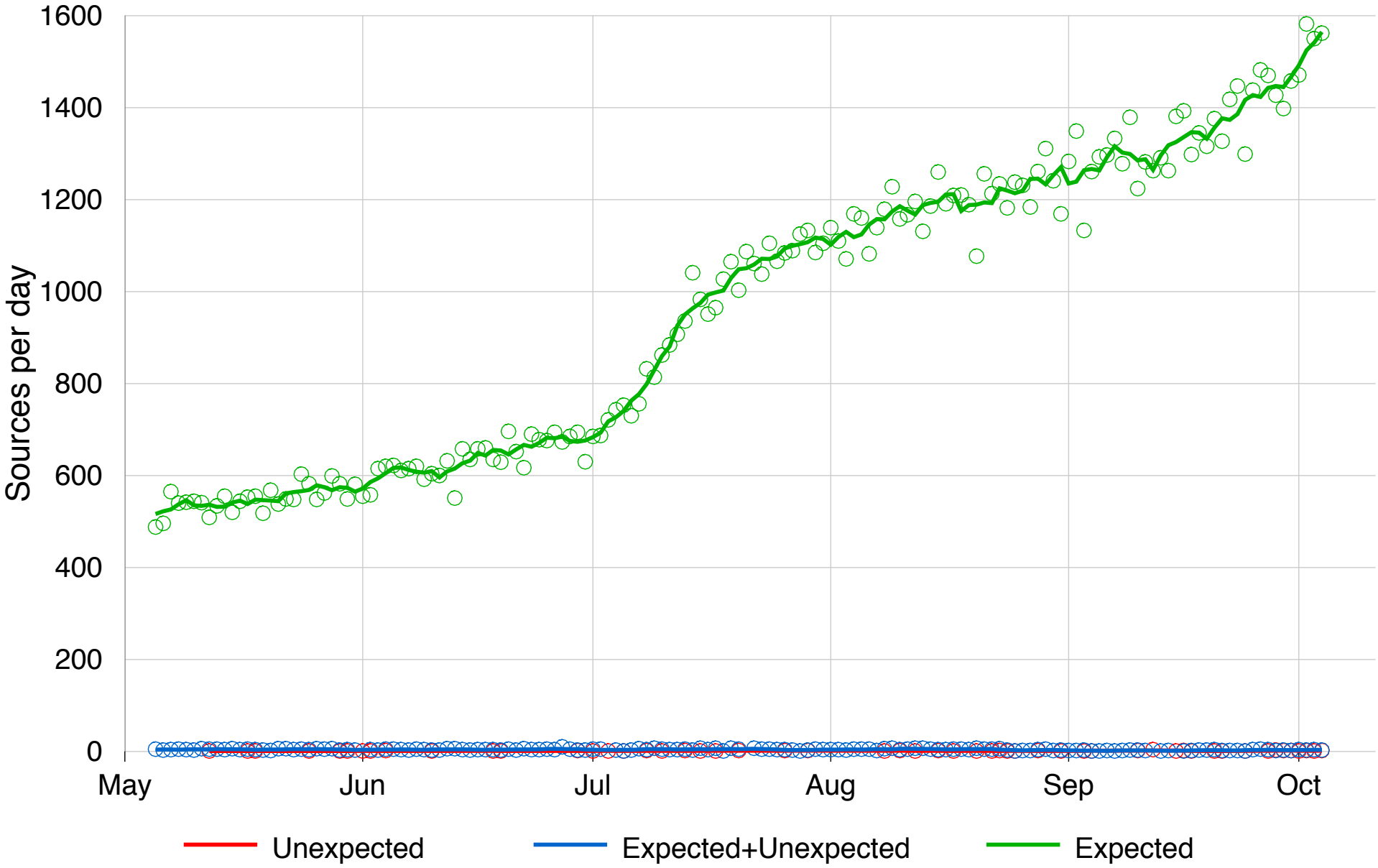

Root Zone Key Tag Signaling –– Number of Sources
Number of Signalers
1600
1400
1200
1000
800
600
400
200
0
May
Jun
Jul
Aug
Sep
Oct
2010
2010/2010+2017
2010+2017


Root Zone Key Tag Signaling –– Number of Sources
Number of Signalers
1600
1400
1200
1000
800
600
400
200
0
May
Jun
Jul
Aug
Sep
Oct
Sources always signaling 2010 TA only.
Sources sometimes signaling 2010 TA and sometimes 2010+2017 TAs
Sources always signaling 2010+2017 TAs only
2010
2010/2010+2017
2010+2017

Non-IANA Key Tags

- How often do we see ”unexpected” key tags?
- Observed 19 key tags for root other than 19036 and 20326.
- From less than 10 distinct source IPs per day.

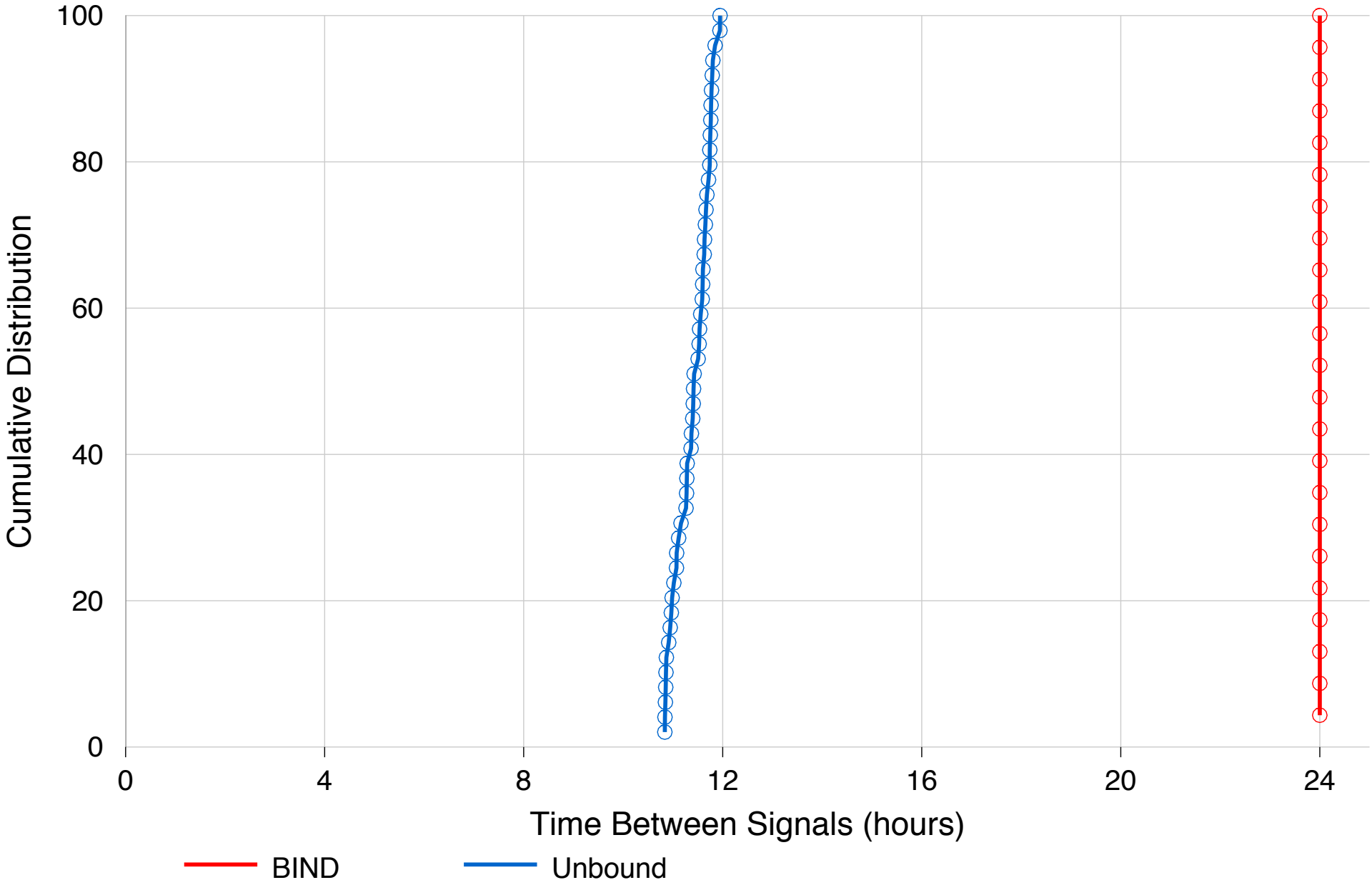
Root Zone Key Tag Signaling –– Unexpected Key Tags

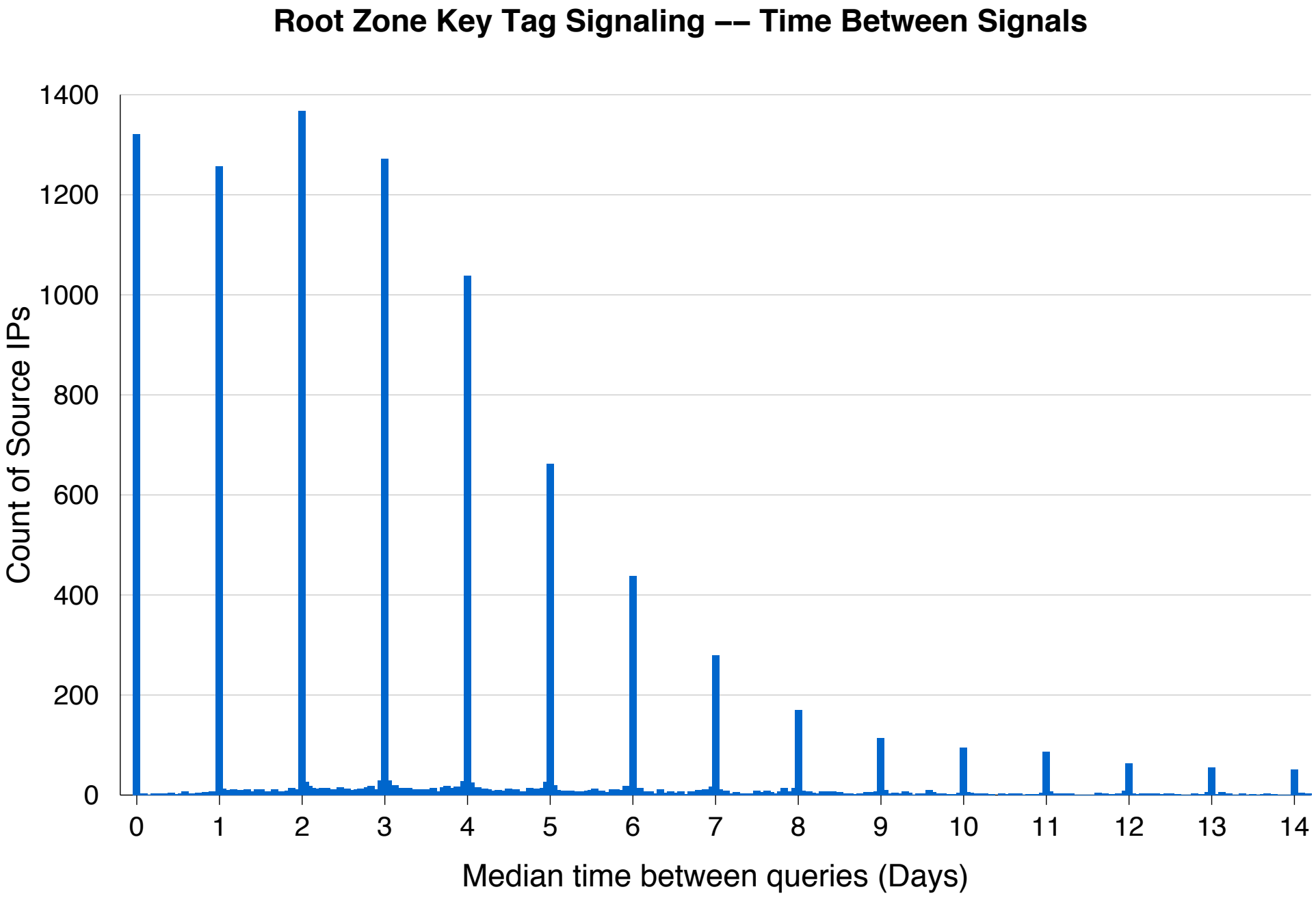


How often do we see key tag queries?

- Do validators report more than once per time-to-live?
- Examine timestamps from self-operated instances of BIND and Unbound.
- Is a partial view useful? e.g., A & J versus all roots?
- Calculate median time between queries from same source.
- Display results as distribution of medians.

Root Zone Key Tag Signaling –– Time Between Signals




Root Zone Key Tag Signaling –– Time Between Signals
Count of Source IPs
1400
1200
1000
800
600
400
200
0
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
Median time between queries (Days)

Conclusions

- Signals from BIND (and Unbound) appear to be of reasonbly good quality.
- Probably a strong selection bias due to newness of the protocol.
- Low level of noise, for now anyway.
- edns-key-tag option may never get deployed.

- ISC, Thank you!
- NLnet Labs, thanks for changing trust-anchor-signaling to ‘yes’ by default.
- Other vendors, please consider implementing RFC 8145.

powered by

VERISIGN™