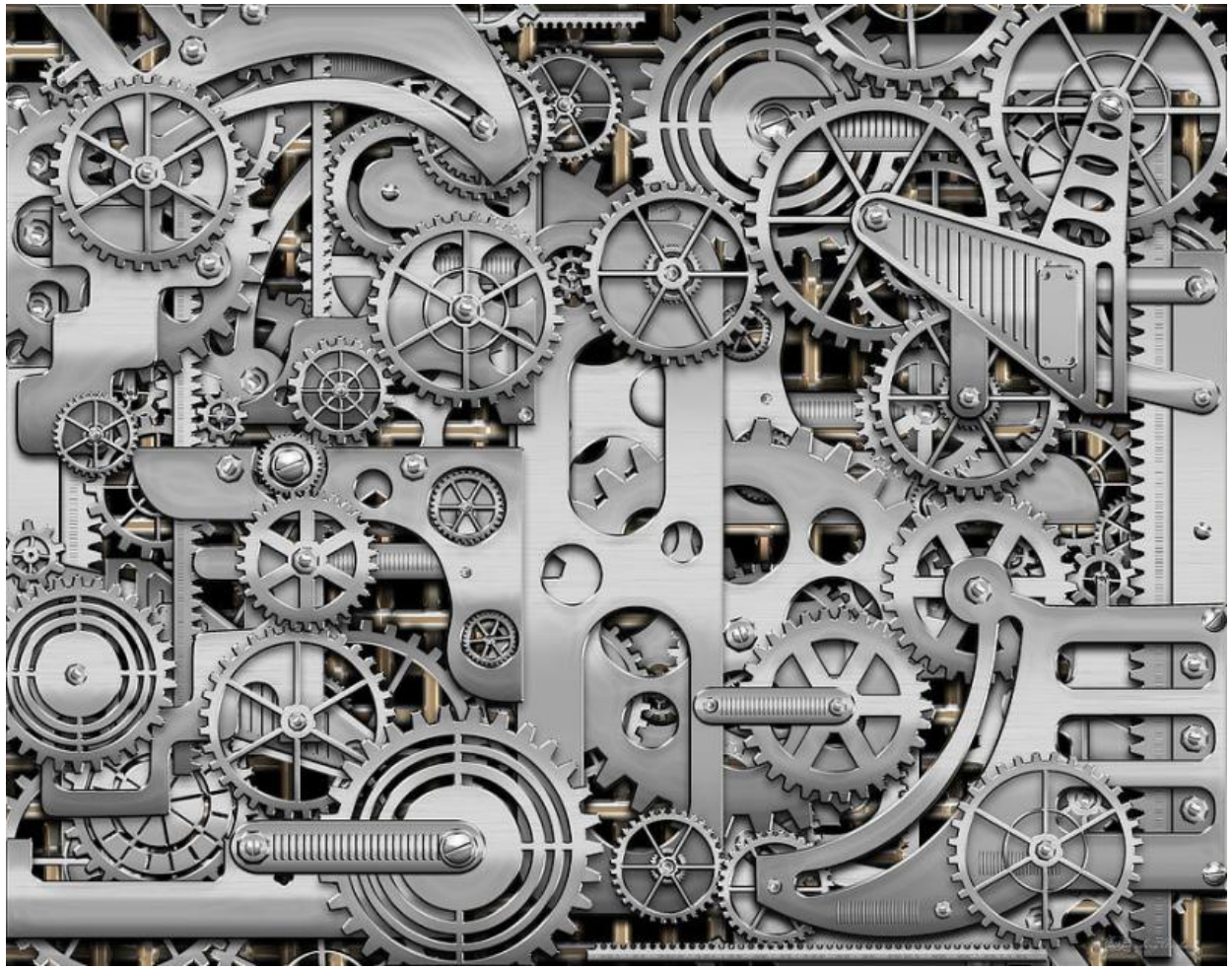
Measuring KSK Roll Readiness

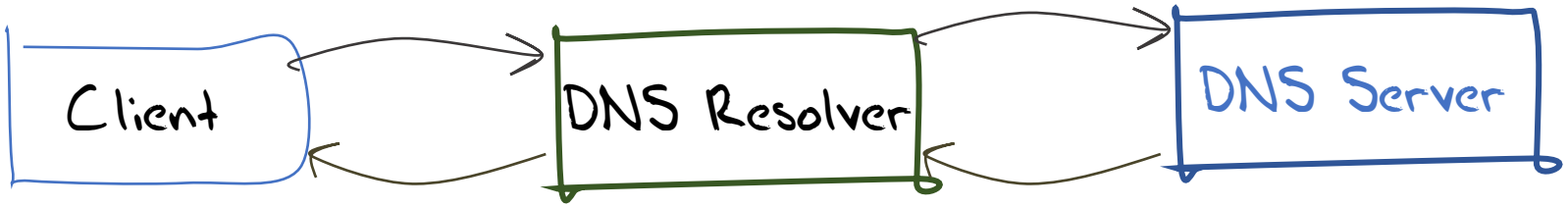
Geoff Huston
APNIC Labs

The DNS may look simple

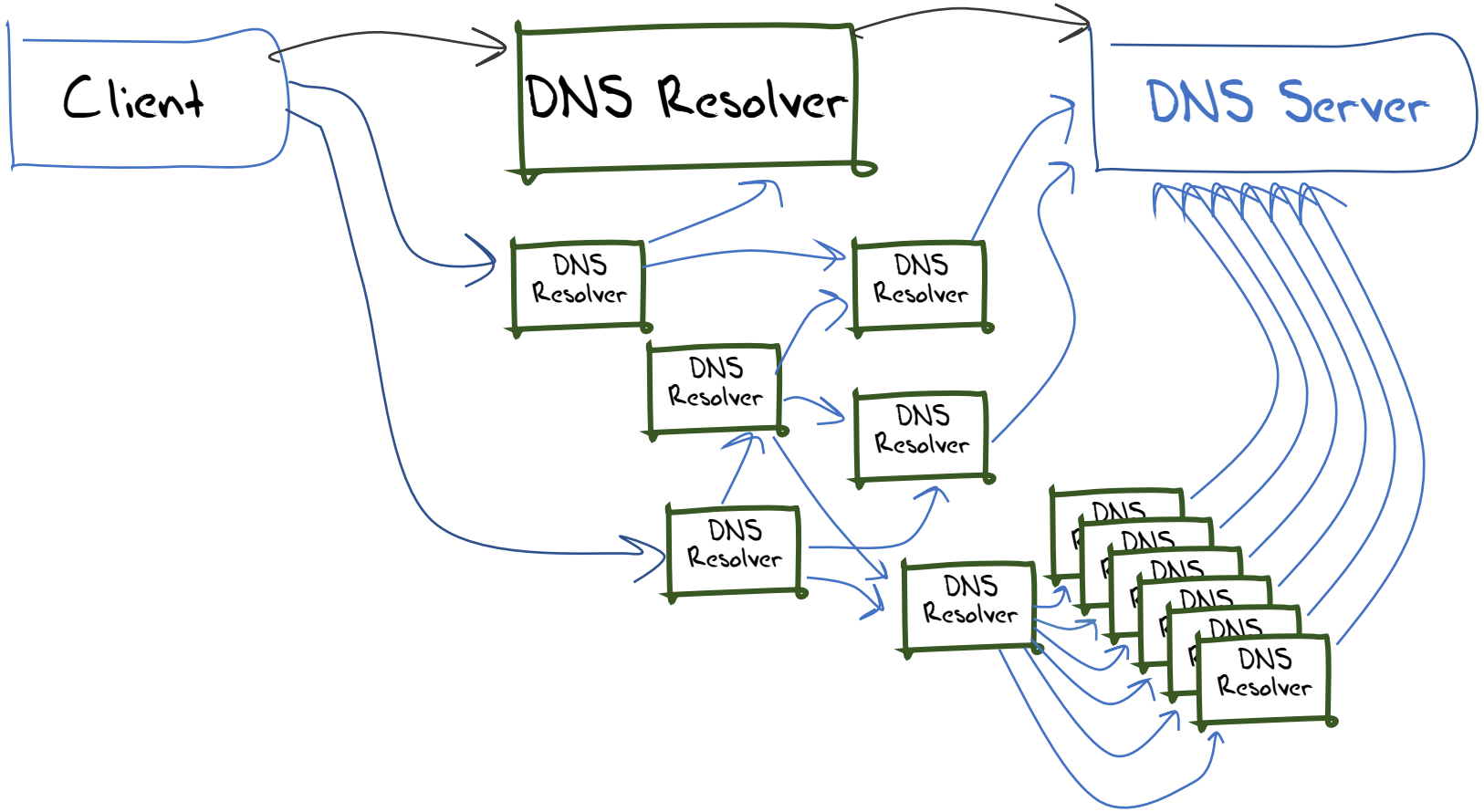
For the DNS looks are very deceiving



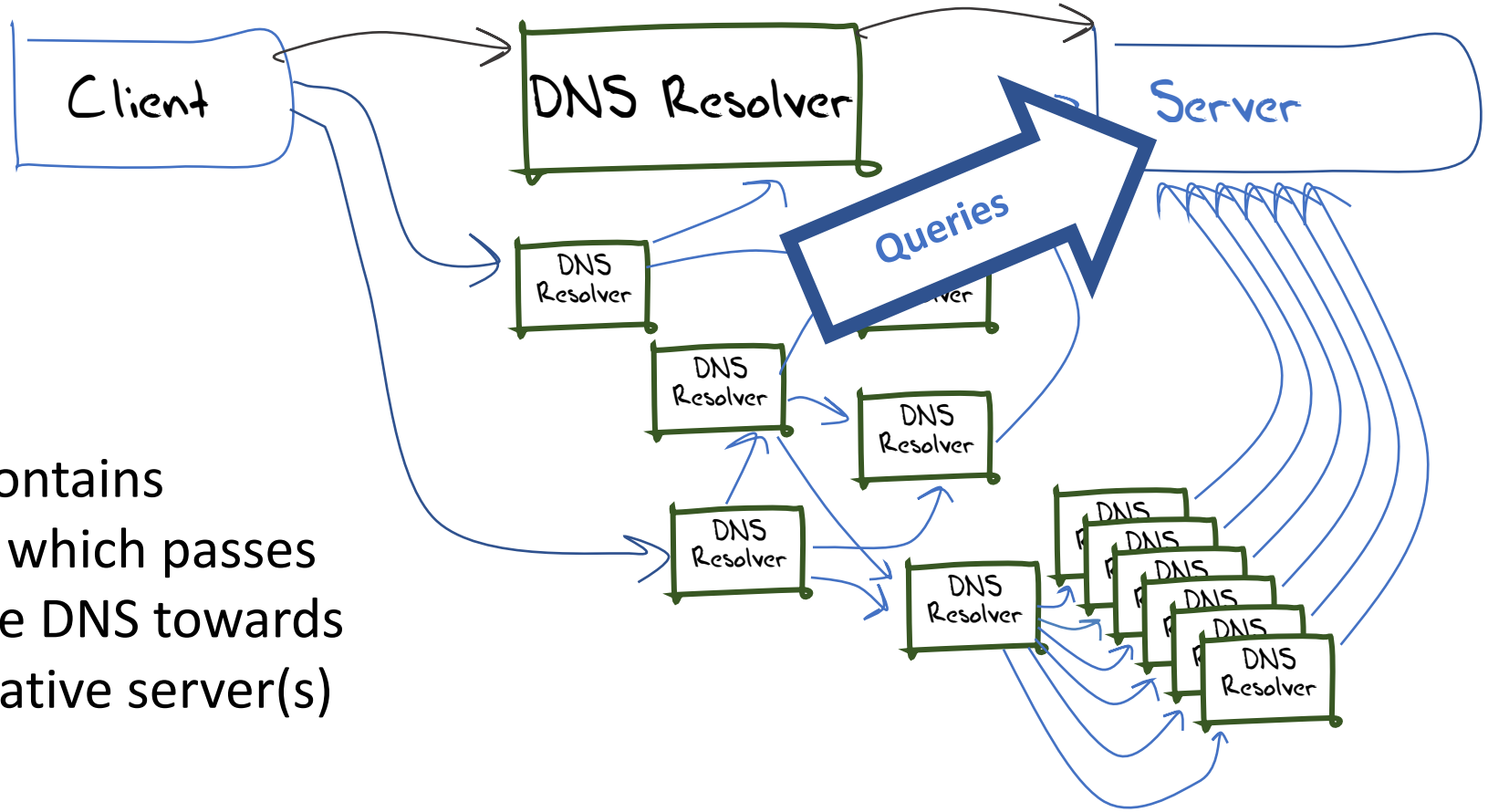
What we would like the DNS to be



What we suspect is more like theDNS

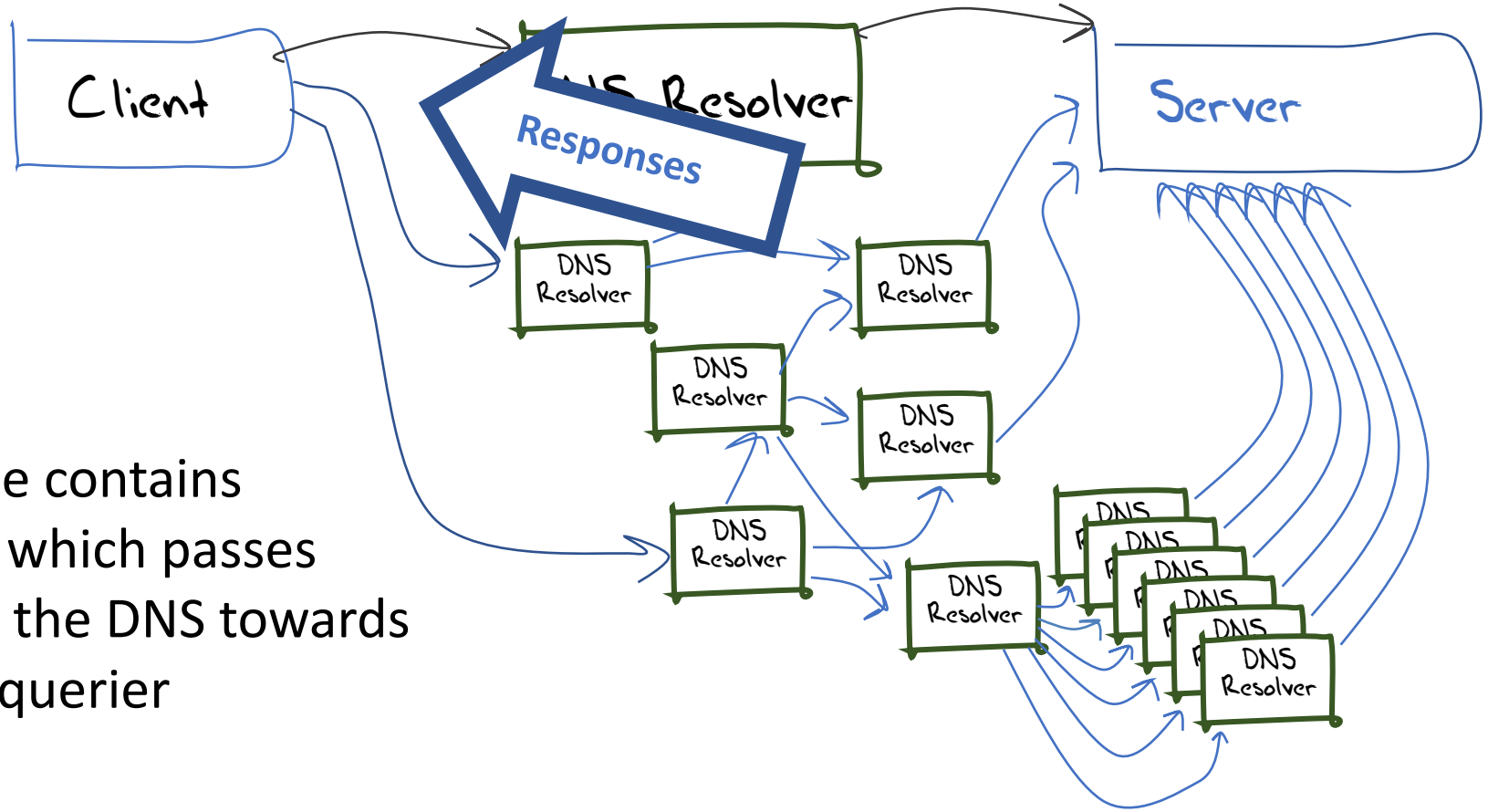


Signalling via Queries



The query contains information which passes inward in the DNS towards the authoritative server(s)

Signalling via Responses



The response contains information which passes backward in the DNS towards the original querier

KSK Roll Measurement Objective

What number of users are at risk of being impacted by the KSK Roll?

- There are two risk elements for resolvers:
  - Unable to receive a 1,414 octet UDP response from the root servers (query for DNSKEY RR from the root zone)
  - Failure to follow RFC5011 key introduction procedure

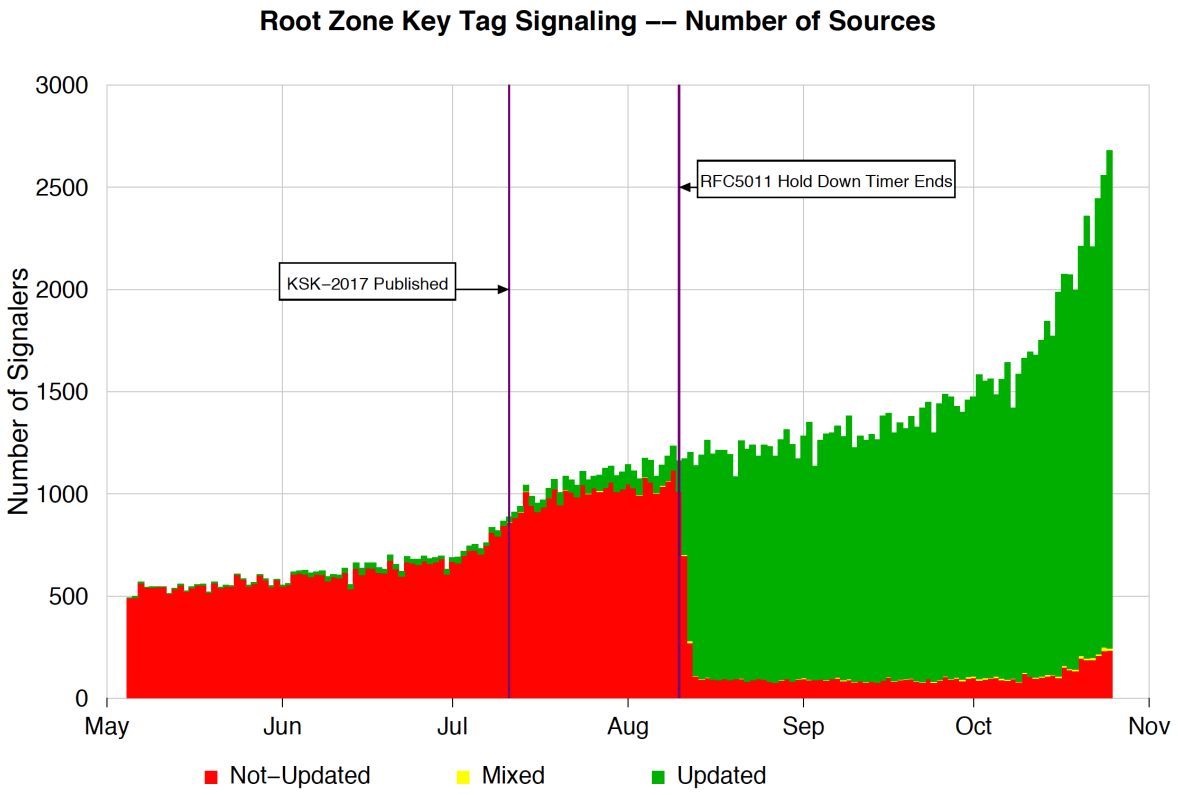
- In either case the resolver outcome is the same: Not loading the incoming trust key into the local trusted key store
- And if the user passes queries **only** to these affected resolvers than the roll will cause a loss of DNS service

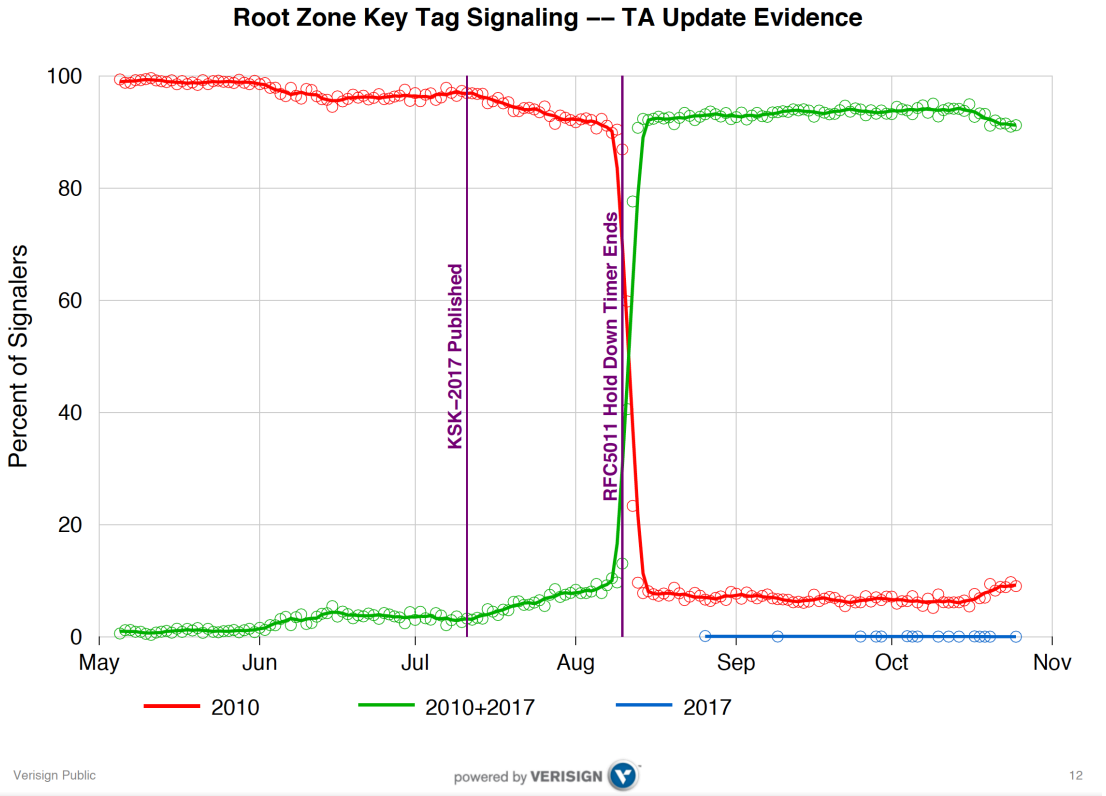
Measuring Resolvers via RFC8145 Signaling

Getting resolvers to report on their local trusted key state

- Resolvers that support the RFC 8145 signal mechanism periodically include the key tag of their locally trusted keys into a query directed towards the root servers

What did we see at (some) roots?





Duane Wessels VeriSign RFC 8145 Signaling Trust Anchor Knowledge In DNS Security Extensions
Presentation to DNSSEC Workshop @ ICANN 60 – 1 Nov 2017
https://schd.ws/hosted_files/icann60abudhabi2017/ea/Duane%20Wessels-VeriSign-RFC%208145-Signaling%20Trust%20Anchor%20Knowledge%20in%20DNS%20Security%20Extensions.pdf

What is this saying?

- Its clear that there is some residual set of resolvers that are signalling that they have not yet learned to trust the new KSK key
- But its not clear if:
  - This is an accurate signal about the state of this resolver
  - This is an accurate signal about the identity of this resolver
  - How many users sit ‘behind’ this resolver
  - Whether these uses rely solely on this resolver, or if they also have alternate resolvers that they can use
  - What proportion of all users are affected

Why?

- Because the DNS does not disclose the antecedents of a query
  - If A forwards a query to B, who queries a Root Server then if the query contains an implicit signal (as in this case) then it appears that B is querying, not A
  - At no time is the user made visible in the referred query
- Because caching
  - If A and B both forward their queries via C, then it may be that one or both of these queries may be answered from C's cache
  - In this case the signal is being suppressed
- Because its actually measuring a cause, not the outcome
  - Its measuring resolvers' uptake of the new KSK, but is not able to measure the user impact of this

User-Side Measurement

Can we devise a DNS query that could reveal the state of the trusted keys of the resolvers back to the user?

- Not within the current parameters of DNSSEC and/or resolver behaviour

User-Side Measurement

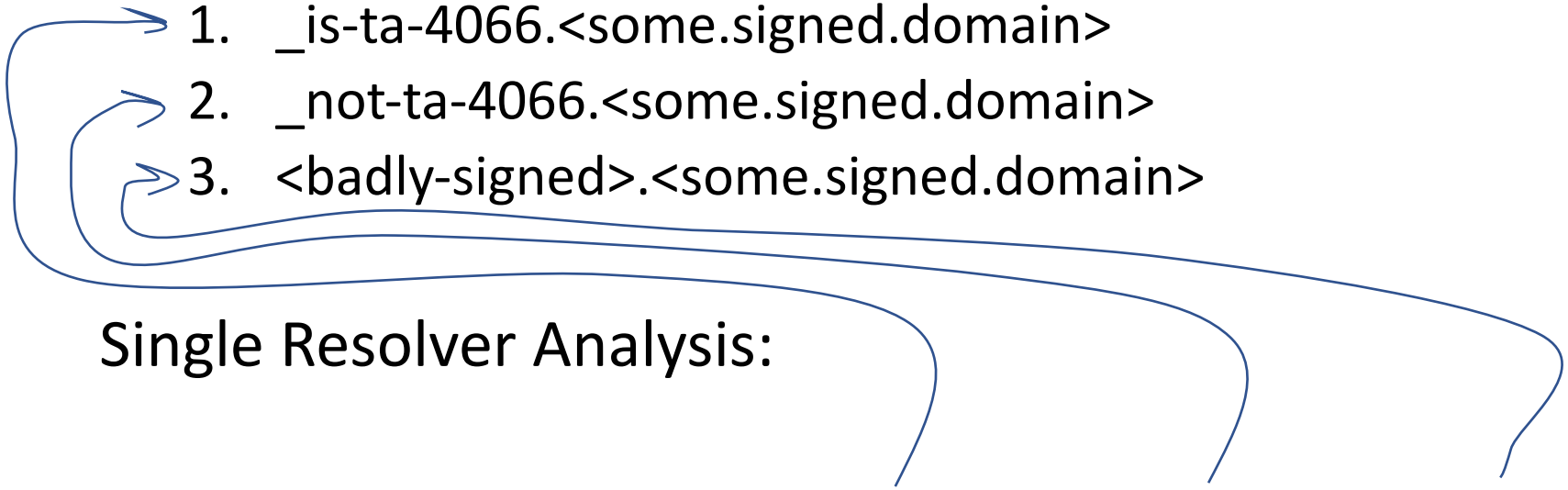
Can we devise a DNS query that could reveal the state of the trusted keys of the resolvers back to the user?

- What if we could change resolver behaviour?
  - Just as RFC8145 required a change in resolver behaviour
- What about a change to the resolver's reporting of validation outcome depending on the resolver's local trusted key state?
  - If a query contains the label "**_is-ta-<key-tag>**" then a validating resolver will report validation failure if the key is NOT in the local trusted key store
  - If a query contains the label "**_not-ta-<key-tag>**" then a validating resolver will report validation failure if the key IS in the local trusted key store

User-Side Resolver Measurement

Three DNS queries:

1. _is-ta-4066.<some.signed.domain>
2. _not-ta-4066.<some.signed.domain>
3. <badly-signed>.<some.signed.domain>



Single Resolver Analysis:

Resolver Behaviour Type	**Query 1**	**Query 2**	**Query 3**
Loaded New KSK	A	SERVFAIL	SERVFAIL
NOT loaded New KSK	SERVFAIL	A	SERVFAIL
Mechanism not supported	A	A	SERVFAIL
Not validating	A	A	A

User-Side DNS Measurement

Multiple Resolver Analysis

A SERVFAIL response will cause the use to repeat they query to other configured resolvers. In a multi-resolver scenario, and where forwarders are used we can still determine if the user will be impacted by the KSK roll

User Impact	Query 1	Query 2	Query 3
OK	A	SERVFAIL	SERVFAIL
NOT OK	SERVFAIL	A	SERVFAIL
UNKNOWN	A	A	SERVFAIL
	SERVFAIL	SERVFAIL	SERVFAIL
NOT Impacted	A	A	A

Measuring User Impact

- Create these tests in a scripted web page and allow users to test the state of their resolvers
- Load these tests into an online ad campaign and use the ad to pass the test to millions of users
  - If the user can resolve Query 1, and SERVFAILs on Query 2 and Query 3 then the user is able to validate using the nominated key as a trusted key
  - If the user SERVFAILS on Query 1, resolves Query 2 and SERVFAILs on Query 3 then the user is unable to validate using the nominated key as a trusted keys
  - Otherwise if the user SERVFAILS on Query 3 then the result is indeterminate

Privacy and Security Considerations

- This test itself does not reveal which resolvers are used by end users in resolving names
- The query itself need not contain any end user identifying material
- The methodology never changes “insecure” to ”authenticated” – it will only change “authenticated” to “insecure” depending on the resolver’s local trusted key state when resolving certain labels
- Anyone can set up a test condition within their delegated part of the DNS
- The results of the test are passed back only to the user in the form of a resolution outcome

A Description of the Mechanism

draft-huston-kskroll-sentinel

[Docs] [txt|pdf] [Tracker] [Email] [Diff1] [Diff2] [Nits]

DNSOP
Internet-Draft
Intended status: Standards Track
Expires: April 29, 2018

G. Huston
J. Damas
APNIC
W. Kumari
Google

October 26, 2017

A Sentinel for Detecting Trusted Keys in DNSSEC
draft-huston-kskroll-sentinel-02.txt

Abstract

The DNS Security Extensions (DNSSEC) were developed to provide origin authentication and integrity protection for DNS data by using digital signatures. These digital signatures can be verified by building a chain of trust starting from a trust anchor and proceeding down to a particular node in the DNS. This document specifies a mechanism that will allow an end user to determine the trusted key state of the resolvers that handle the user's DNS queries.

Thanks!