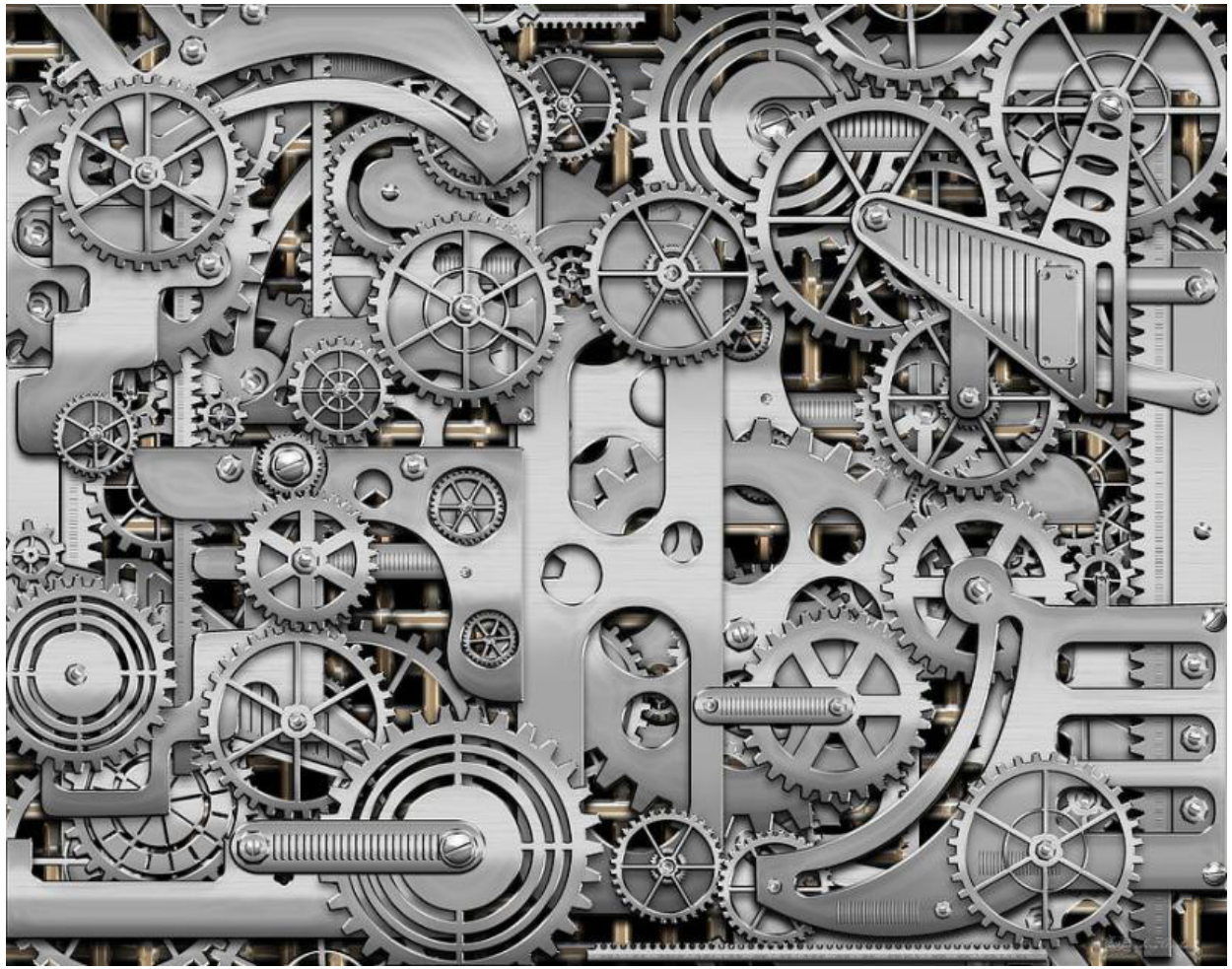
That KSK Roll

Geoff Huston
APNIC Labs

The DNS may look simple

But with the DNS, looks are very deceiving



So lets talk DNSSEC

- DNSSEC introduces digital signatures into the DNS
- It allows a DNS resolver to validate the information it receives to ensure that however it got to learn it, the information is:
  - Valid
  - Up to Date

- It can prevent various forms of attack on the DNS
- But more importantly it can allow us to publish information in the DNS and have clients authenticate that the information has not been tampered with in any way

Why DNSSEC?

- To improve the robustness of the DNS

  Because we can inject false information into the DNS and mislead users

- We could improve the robustness of a secured web

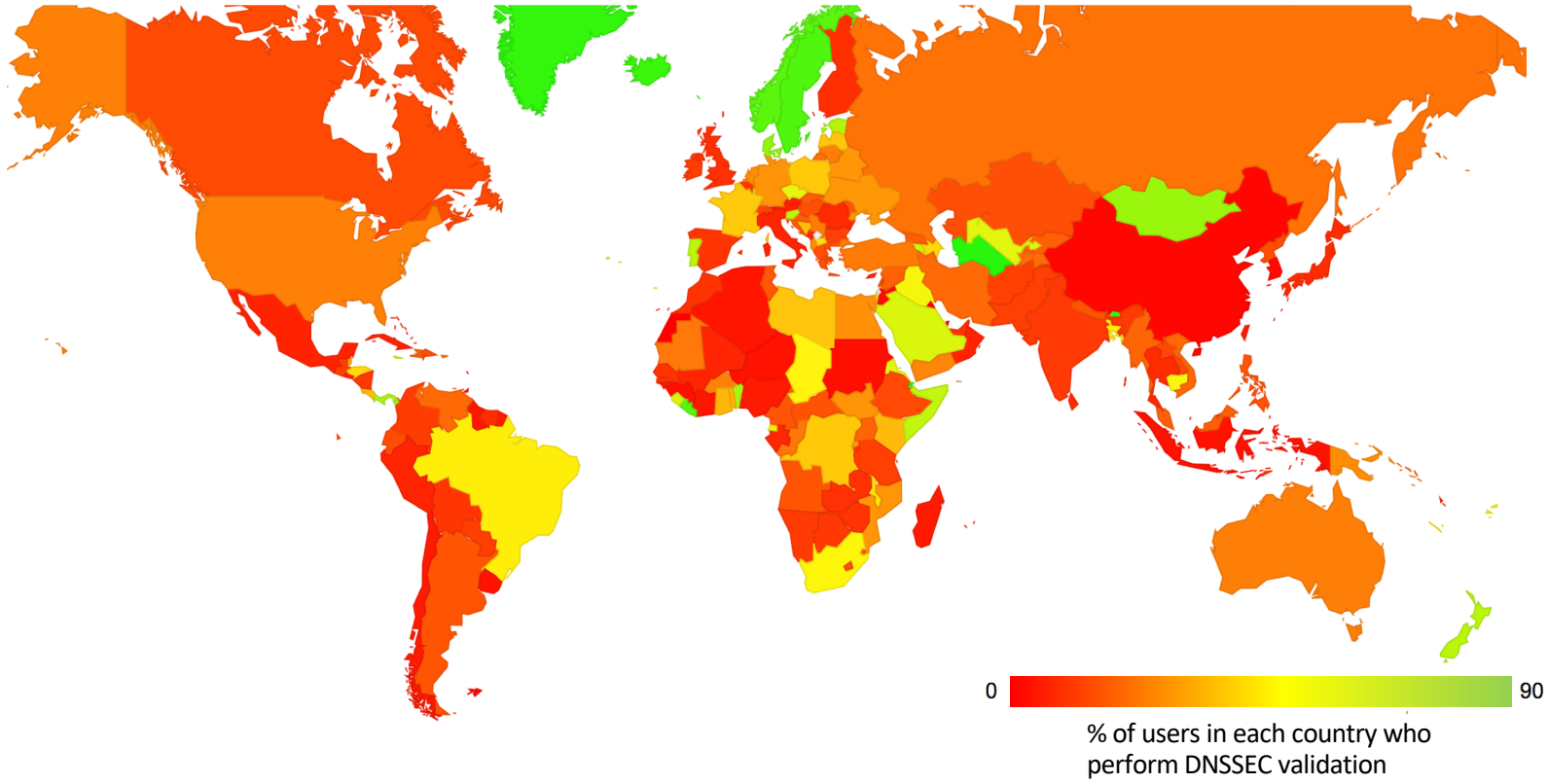
  Because authentic information in the DNS can help in propping up the WebPKI model with CA pinning information

- To improve access to robust security

  Because we'd like to make some progress towards an environment where accessible, robust security in accessing services can be provided

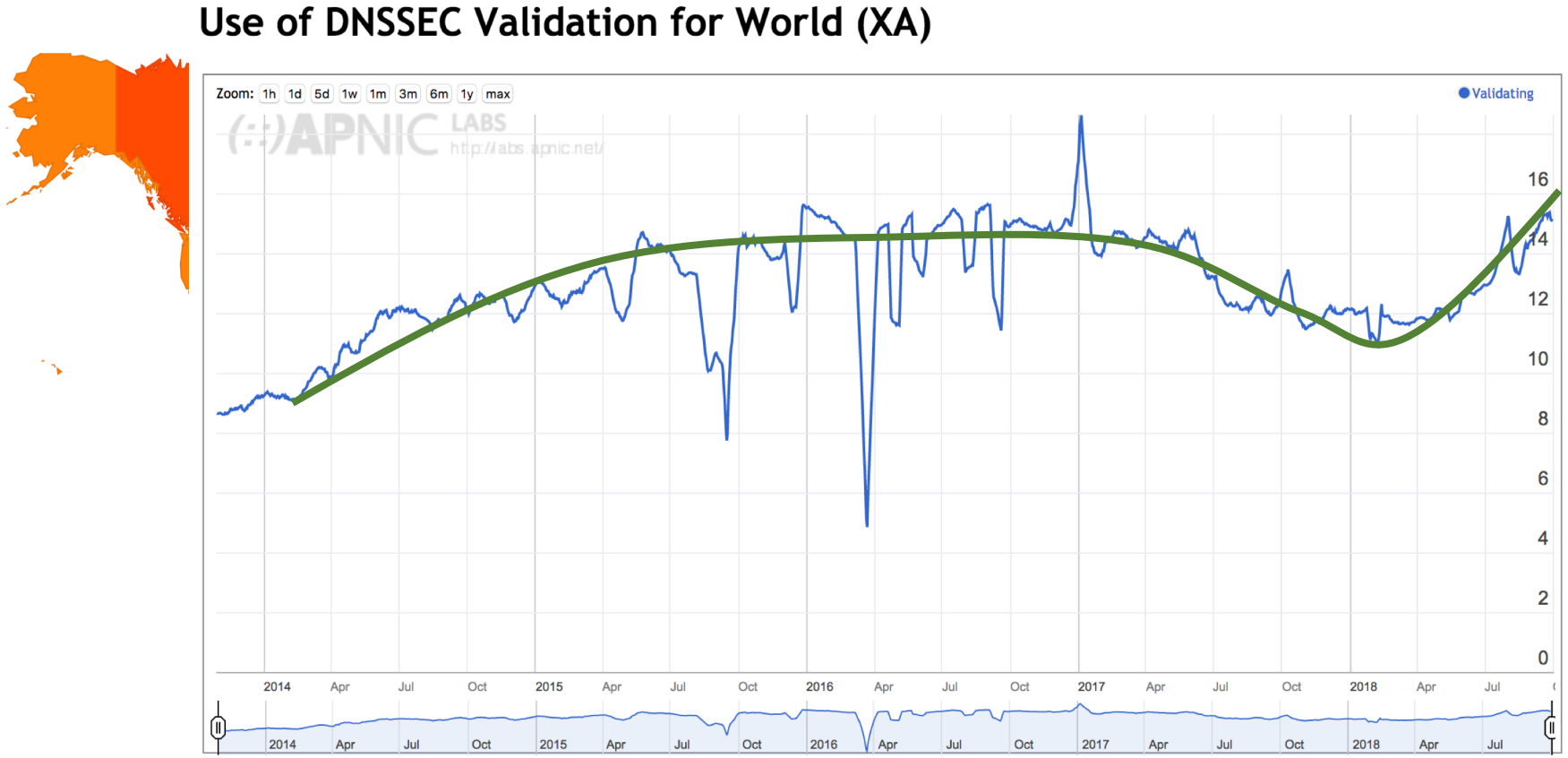
Who uses DNSSEC?

Today around 1 in 7 users will not resolve a DNSSEC-signed domain name if the signature cannot be validated



Who uses DNSSEC?

Today around 1 in 7 users will not resolve a DNSSEC-signed domain name if the signature cannot be validated



Trust in DNSSEC

DNSSEC uses a single point of trust – the Key Signing Key of the root zone

- The DNSKEY for the root zone is signed by the KSK for the root zone
- This key signs a DNSKEY entry that includes a Zone Signing Key – ZSK - that is used to sign every entry in the root zone
- The zone includes a DS entry signed by the root zone ZSK for the .net delegation that is the hash of the KSK for .net

  - The DNSKEY for .net is signed by the KSK for .net
  - The DNSKEY entry for .net includes a ZSK used to sign every entry in the .net zone
  - The zone includes a DS entry signed by the .net zone ZSK for the potaroo.net delegation that is the hash of the KSK for .potaroo.net

    - The DNSKEY for potaroo.net is signed by the KSK for potaroo.net
    - The DNSKEY entry for potaroo .net includes a ZSK used to sign every entry in the potaroo.net zone
    - The entry for www.potaroo.net is signed by the ZSK for potaroo.net

Rolling keys in DNSSEC

- Rolling a key can be easy in DNSSEC for most keys:
  - Tell your ”parent” about the new key

    - wait -
  - Sign the products with the new key

    - wait -
  - Withdraw the old key

Rolling keys in DNSSEC

- Rolling a key can be easy in D… most keys:
  - Tell your ”parent” a… key
    - wait -
  - Sign … with the new key
  - …ithdraw the old key

But the Root Zone has no parent, so this approach doesn't work for the KSK

Rolling Keys

- No key is eternal:
  - Digital cryptography is based on difficult problems, not impossible problems! As computing capabilities change, what is unfeasible to solve changes
  - The best crafted plans can fail – key compromise is always a possibility
  - Key storage systems can fail – inability to access the key makes the key useless
- If we need to instil a discipline of changing keys in relying party systems then we need to change the key from time to time
- Is it better to gain experience in regularly scheduled key rolls than react with ad hoc measures when it's forced upon us?

The Mechanics of Rolling the KSK

1. ”Introduce” the new KSK to the world
2. Use the Root Zone DNSKEY record to publish a new KSK

   - wait - → See RFC 5011 – wait for at least 30 days
3. Sign the Root Zone DNSKEY record with the new KSK
4. Remove the old KSK from the Root Zone DNSKEY record

   - wait -
5. Revoke the old KSK in the Root Zone DNSKEY record

The Mechanics of Rolling the KSK

1. ”Introduce” the new KSK to the world
2. Use the Root Zone DNSKEY record to publish a new KSK

   - wait -
3. Sign the Root Zone DNSKEY record with the new KSK
4. Remove the old KSK from the Root Zone DNSKEY record

   - wait -
5. Revoke the old KSK in the Root Zone DNSKEY record

We are here

October 11

Will this work?

We don’t really know!

- If everyone built code strictly according to current specs

And

- All specs were clear, unambiguous and functional

And

- We could build robust accurate code

And

- The Internet worked all of the time

And

- The stars are aligned in our favour

Then everything will be fine!

But

We really don’t know!

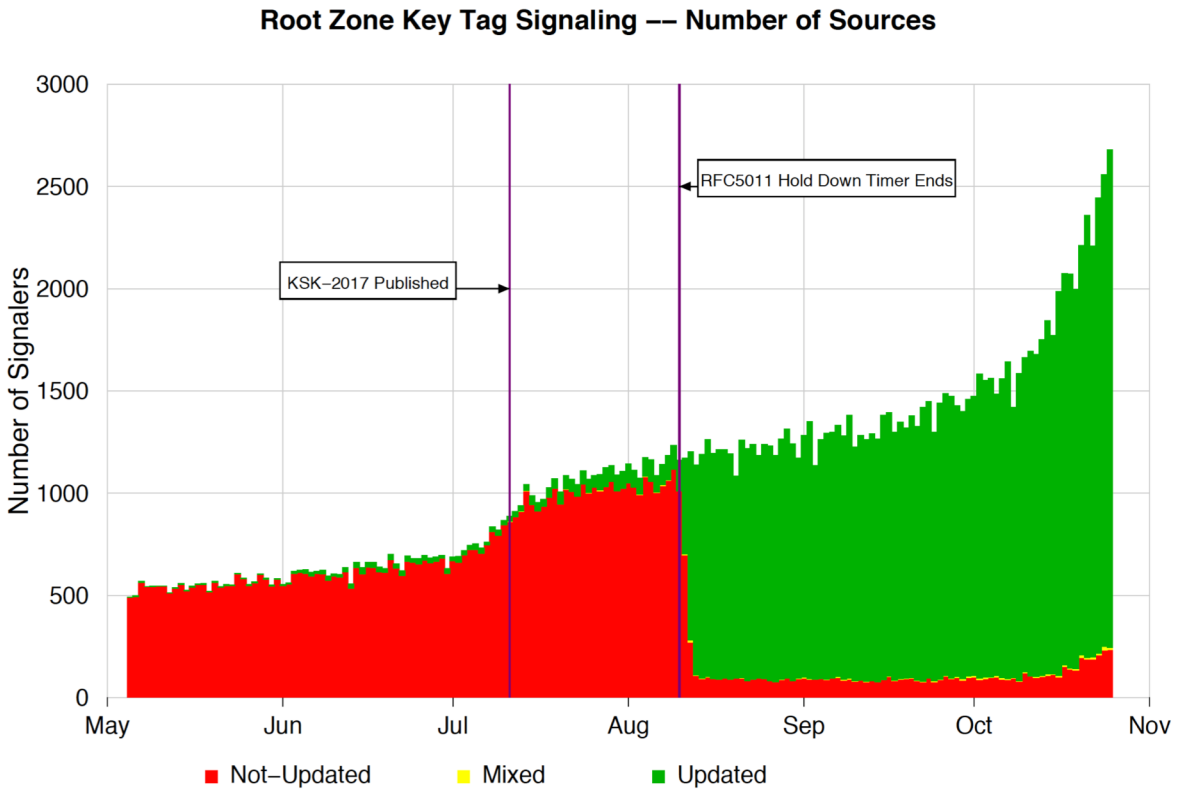
Is there a way we might peek into the DNS and see just how ready we are for a KSK roll?

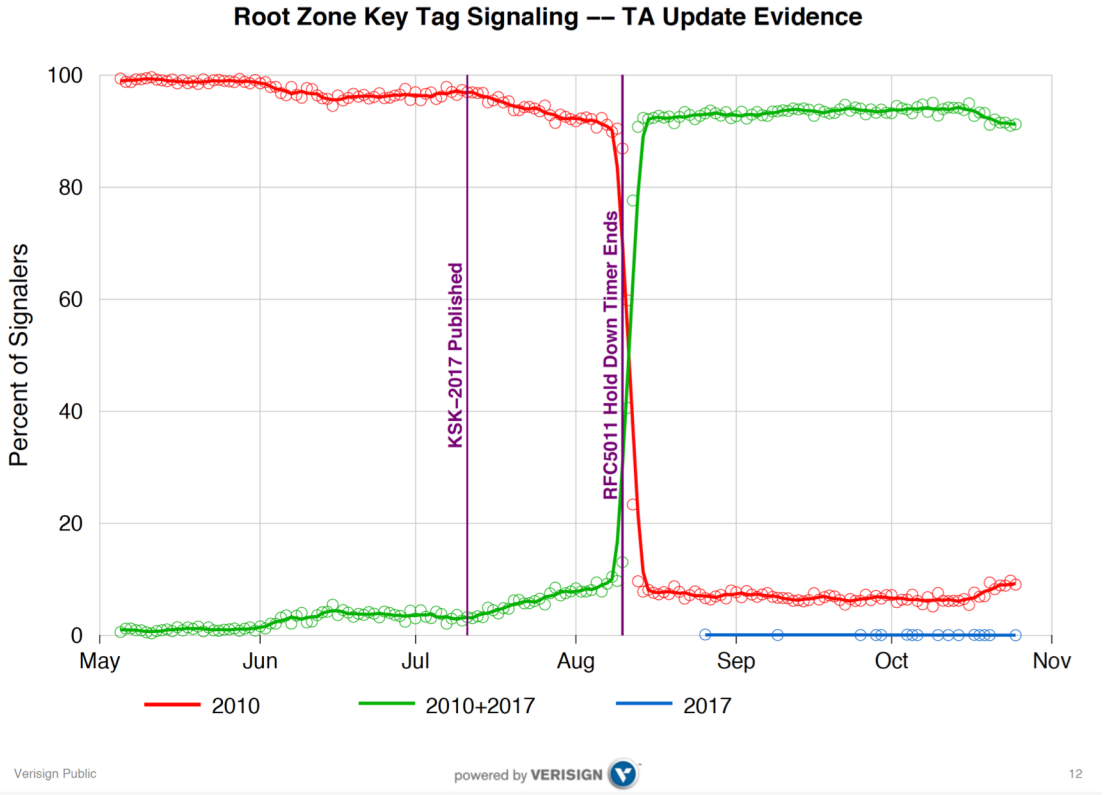
Measuring Resolvers via RFC8145 Signaling

Getting resolvers to report on their local trusted key state

- A change to resolver behavior that requires deployment of new resolver code
- 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 RFC 8145 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 it’s not all that helpful as a signal
  - Is this is an accurate signal about the state of this resolver?
  - Is 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 do *tracequery*
  - Forwarding a query onward in the DNS contains no ‘sticky’ information about the original query
  - At no time is the user exposed 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?

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 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 **"root-key-sentinel-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 **"root-key-sentinel-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. root-key-sentinel-is-ta-20326.<unique-label>.<some.signed.domain>
2. root-key-sentinel-not-ta-20236.<unique-label>.<some.signed.domain>
3. <badly-signed>. <unique-label>.<some.signed.domain>

Analysis:

Resolver Behaviour Type	**Query 1**	**Query 2**	**Query 3**
Loaded KSK-2017	A	SERVFAIL	SERVFAIL
NOT Loaded KSK-2017	SERVFAIL	A	SERVFAIL
Mechanism not supported	A	A	SERVFAIL
Not validating	A	A	A

Measuring User Impact

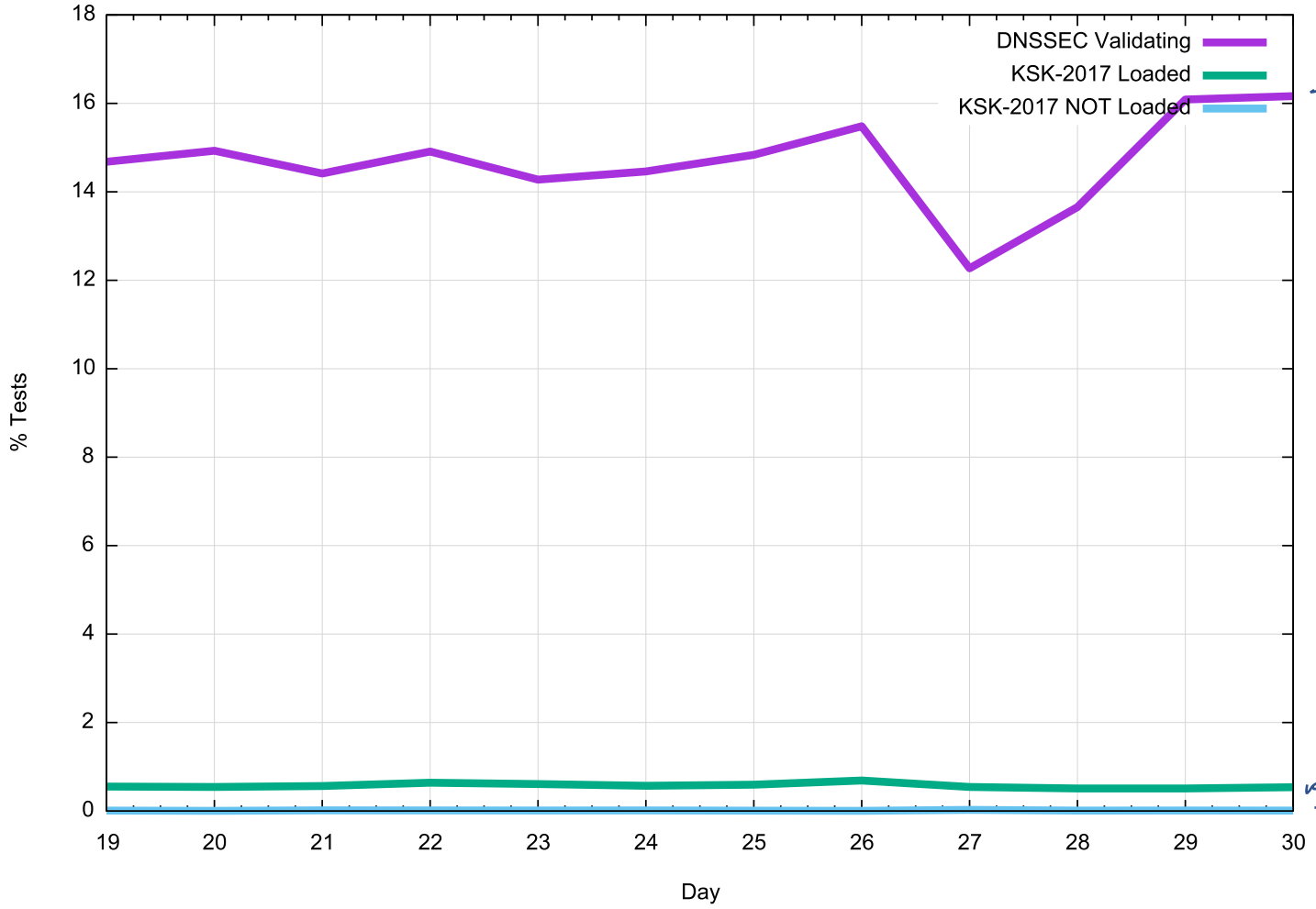
- Load these DNS tests into an online ad campaign and use the ad to perform this measurement with at least hundred million users across the Internet before the key rolls

Are we ready?

The KSK rolls on 11 October 2018

And we've been measuring for the past week

Results so far



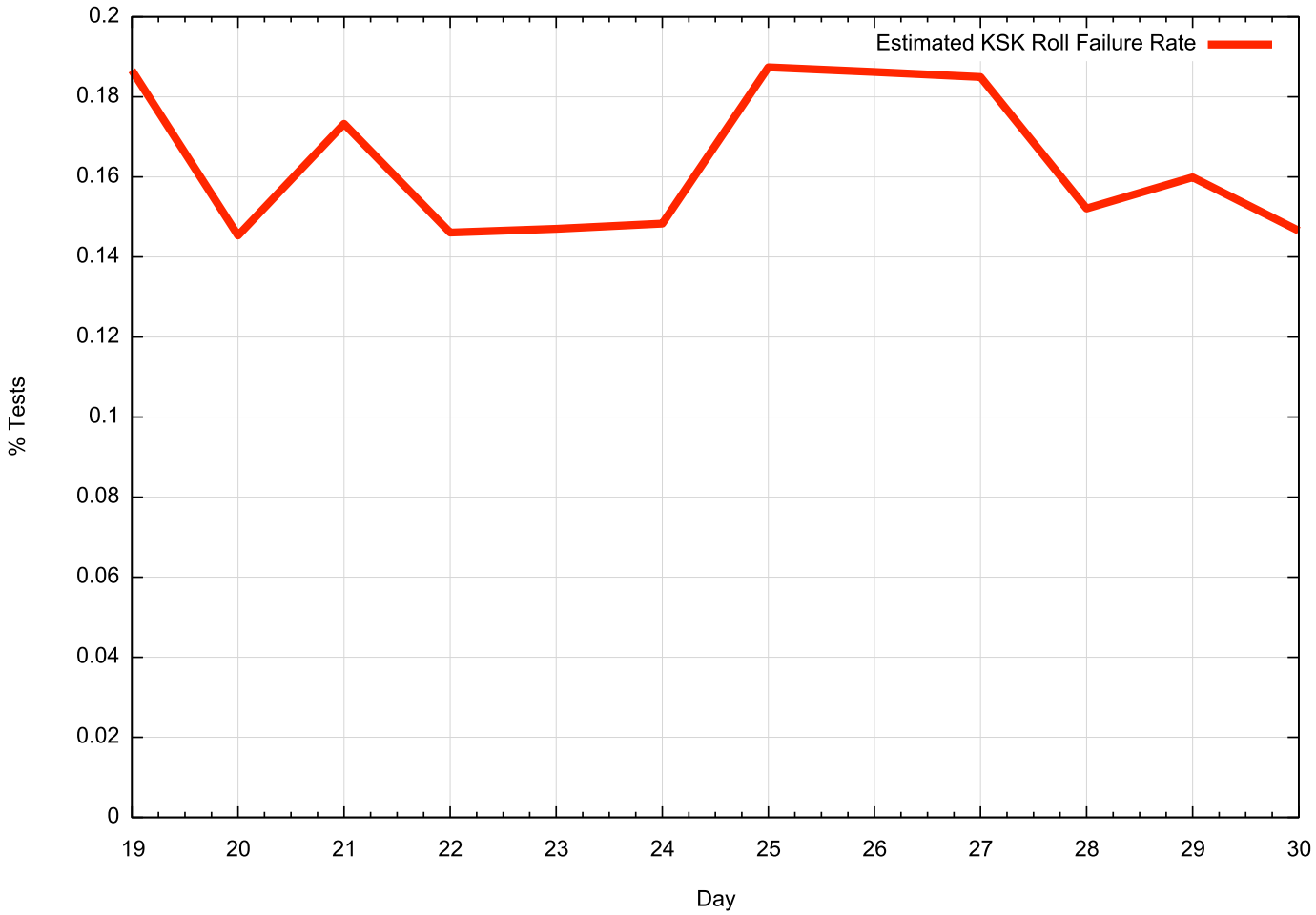
16% of users use DNSSEC-validating resolvers

15% of users do not report their KSK trust-state

0.5% of users report KSK-2017 loaded

0.005% of users report KSK-2017 NOT loaded

Possibly Affected Users



Between 0.1% to 0.2% of users are reporting that their resolvers have not loaded KSK-2017 as a trust anchor

The measurement has many uncertainties and many sources of noise so this is an upper bound of the pool of users who may encounter DNS failure due to to the KSK roll

There is still uncertainty in this measurement

- This is a recent change to resolver behaviour – only a few resolvers respond to the key test queries!
- Not all resolvers will pre-provision KSK-2017 using RFC 5011 automated trust mechanisms – they may elect to load the new trust anchor at the time of the roll
  - And we cannot measure the difference between a resolver that has a broken implementation of RFC5011 and a resolver that is being managed manually
- We cannot readily measure the properties of individual resolvers
  - We can derive an estimate of users that may be impacted, but it’s not so easy to figure out which resolvers are responsible

What to expect with the KSK roll

- To the extent that the DNS is able to be measured, it appears that most users will be unaffected by the planned roll of the KSK on October 11
- **But your helpdesk should be aware that we a user reports that their Internet is dead in the 48 hours after October 11 then a DNSSEC trusted key failure for the user is a possible cause.**

Thanks!