

KSK Sentinel

draft-ietf-dnsop-kskroll-sentinel

Geoff Huston
Joao Silva Damas
Warren Kumari

What's the problem?

- We ~~need~~ want to roll the DNSSEC trust-anchor (KSK)
- Users with a resolver that doesn't have the new KSK break; **everything** looks BOGUS
- We have no way of measuring deployment, and so don't know who will break

Wait! RFC8145?!

PROPOSED STANDARD	
Internet Engineering Task Force (IETF) Request for Comments: 8145 Category: Standards Track ISSN: 2070-1721	D. Wessels Verisign W. Kumari  Google P. Hoffman ICANN April 2017
Signaling Trust Anchor Knowledge in DNS Security Extensions (DNSSEC)	
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	

- Sadly, no.
- This provides reporting from **resolver**

I have a validating resolver in my basement...

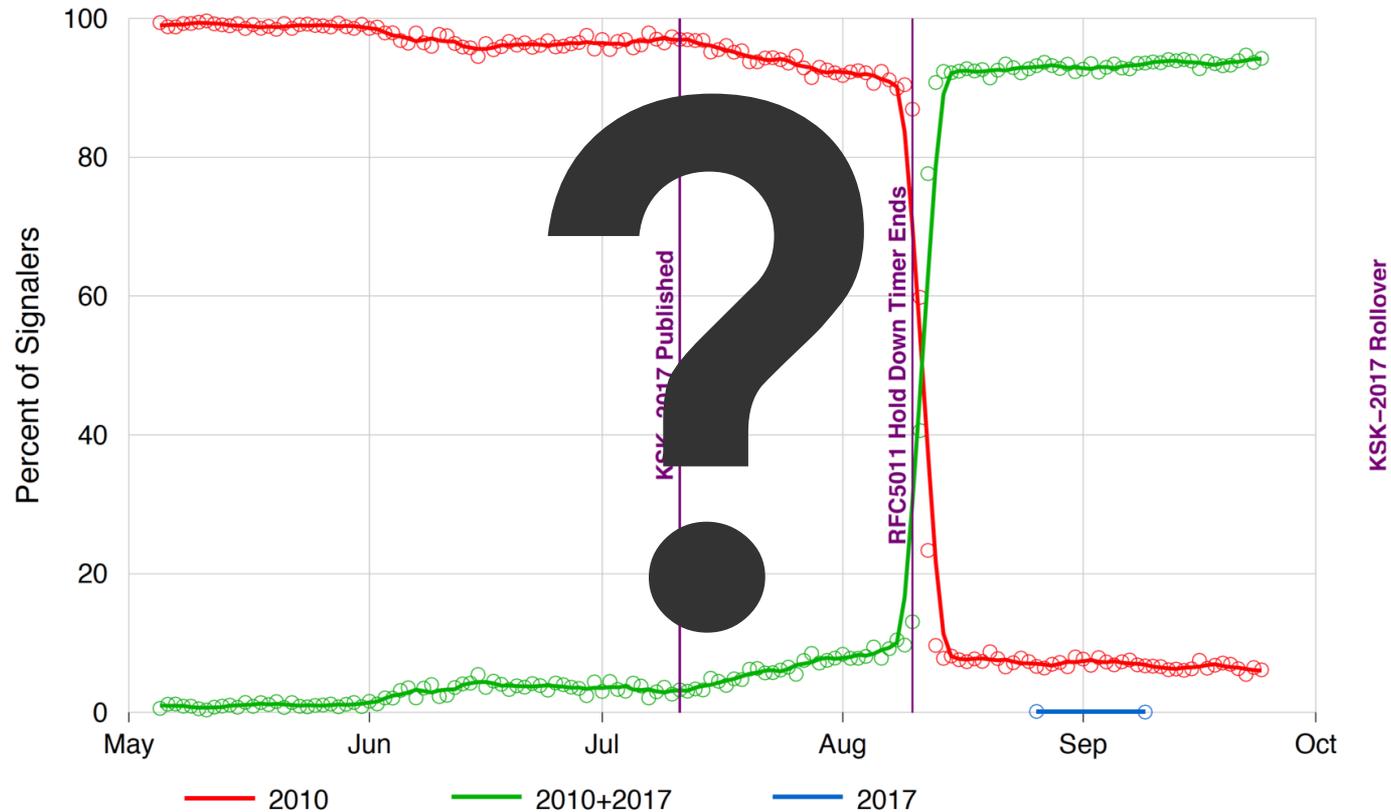
it doesn't have the new key :-)

but no-one is using it :-)

If a resolver falls in the forest, but no-one is using it, does it matter?!

Pretty_graphs!

Root Zone Key Tag Signaling -- TA Update Evidence



Sentinel

1. Requires a (simple) resolver update
2. Allows anyone to set up a measurement service
3. Exposes the result to the users

The change

Just before sending the **response** (after resolution, validation):

- `kskroll-sentinel-is-ta-[key].something?`
 - If have the key, reply normally, else SERVFAIL
- `kskroll-sentinel-not-ta-[key].something?`
 - If do NOT have the key, reply normally, else SERVFAIL

Example

- I'm a validating resolver. I support sentinel.
- I have the new KSK (20326)
- I get a query for `invalid.example.com`
 - It fails DNSSEC validation - SERVFAIL
- I get a query for `kskroll-sentinel-is-ta-20326.example.com`
 - I have (and am using) KeyID 20326
 - answer normally
- I get a query for `kskroll-sentinel-not-ta-20326.example.com`
 - I do have (and am using) KeyID 20326
 - send SERVFAIL



Yawn. So what?!

```
1  <html>
2    <body>
3      <h1>KSK Roll Test</h1>
4      <p>
5        Fish: 
6        Kitten: 
7        Puppy: 
8      </p>
9    </body>
10 </html>
```

Do you see:

- Fish? Not validating, key-roll doesn't affect you.
- Kitten and Puppy? Legacy, we cannot tell.
- Kitten? You have the new key, you'll be fine.
- Puppy? **DANGER!** You only have the old key.

Srsly? Kittens?!

Sadly, no...

```
1 <html>
2 <head>
3   <script type="text/javascript"
4     src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
5 </head>
6 <body>
7   <h1>Sentinel KSK Test</h1>
8   <p hidden>
9     
10    
11    
12  </p>
13
14  <p>
15    <span id="sentinel"></span>
16  </p>
17
18  <script type="text/javascript">
19    var invalid=true, is_ta=true, not_ta=true, result="Testing failed...";
20
21    $('#img_invalid').error(function(){invalid=false});
22    $('#img_is_ta').error(function(){is_ta=false});
23    $('#img_not_ta').error(function(){not_ta=false});
24
25    window.addEventListener('load', function(){
26      switch (true) {
27        case invalid===true:
28          result='No DNSSEC validation, you will be fine..'; break;
29        case (is_ta===true && not_ta===true):
30          result='Legacy resolver, cannot determine your fate!'; break;
31        case (is_ta===true):
32          result='WARNING!: You do not have the new KSK.'; break;
33        case (not_ta===true):
34          result='Congratulations, you have the new key. You will be fine.'; break;
35      }
36      $('#sentinel').text(result);
37    });
38  </script>
39 </body>
40 </html>
```

Srsly? Kittens?!

Still no... :-)

Demo: <http://www.ksk-test.net>:

Sentinel KSK Test

tl;dr: **You are using a legacy resolver, we cannot determine your fate!**

This page uses the methods described in [A Sentinel for Detecting Trusted Keys in DNSSEC](#) to determine if the resolvers that you are using will survive the upcoming KSK roll. You should really read the document, but the 50'000ft view is that it attempts to load resources from 3 names:

- ["http://invalid.ksk-test.net/invalid.gif"](http://invalid.ksk-test.net/invalid.gif)
- ["http://kskroll-sentinel-is-ta-20236.ksk-test.net/is-ta.gif"](http://kskroll-sentinel-is-ta-20236.ksk-test.net/is-ta.gif)
- ["http://kskroll-sentinel-not-ta-20236.ksk-test.net/not-ta.gif"](http://kskroll-sentinel-not-ta-20236.ksk-test.net/not-ta.gif)

It then uses some simple logic to tell what your fate will be after the KSK roll:

1. If you are **not** using a validating resolver, you will be able to load the *invalid* record.
2. If you are using a validating resolver which **does not** understand this new mechanism you will be able to load both of the sentinel records: *kskroll-sentinel-is-ta-20236* and *kskroll-sentinel-not-ta-20236*.
3. If you are using a resolver that supports this mechanism you will only be able to load one of the two sentinel records - which one tells you how you will fare in the rollover.

When running the above test, you:

- were **NOT** able to fetch the "invalid" record
- were able to fetch the "kskroll-sentinel-is-ta-20236" record
- were able to fetch the "kskroll-sentinel-not-ta-20236" record

Questions?

