KSK Sentinel

draft-ietf-dnsop-kskroll-sentinel

Geoff Huston

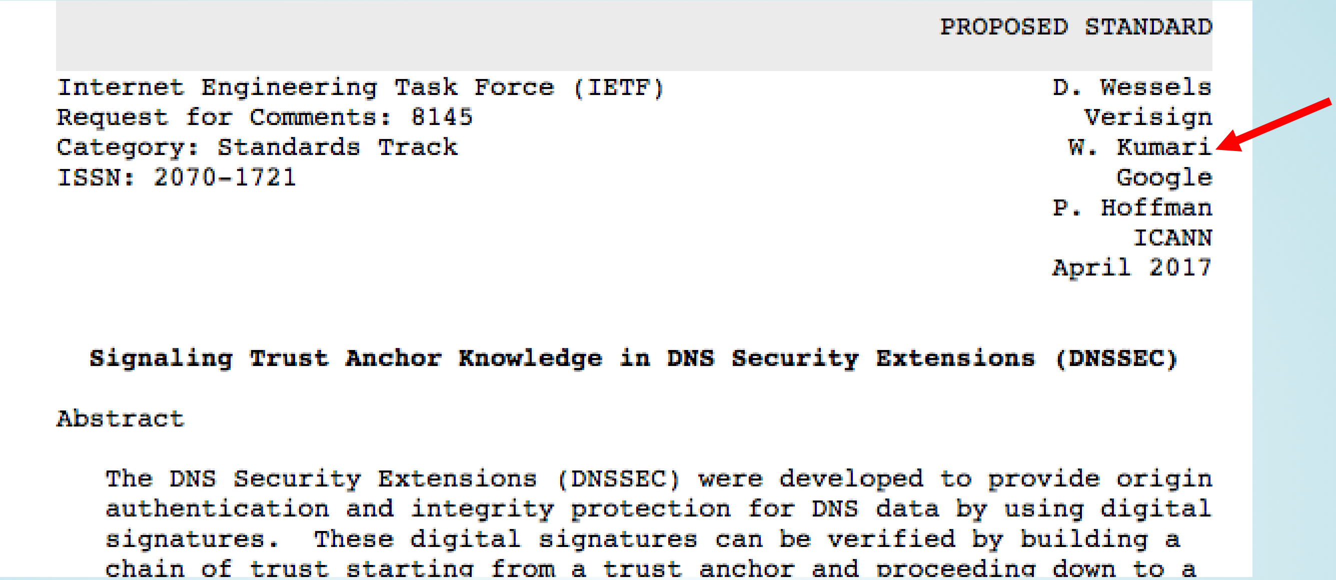
Joao Silva Damas

Warren Kumari

What's the problem?

- We ~~need~~ want to roll the DNSSEC trust-anchor (KSK)
- Users with a validating 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 (and how many!) will break

Wait! RFC8145?!



```
                                                       PROPOSED STANDARD
Internet Engineering Task Force (IETF)                        D. Wessels
Request for Comments: 8145                                      Verisign
Category: Standards Track                                      W. Kumari
ISSN: 2070-1721                                                   Google
                                                              P. Hoffman
                                                                   ICANN
                                                              April 2017
```

Signaling Trust Anchor Knowledge in DNS Security Extensions (DNSSEC)

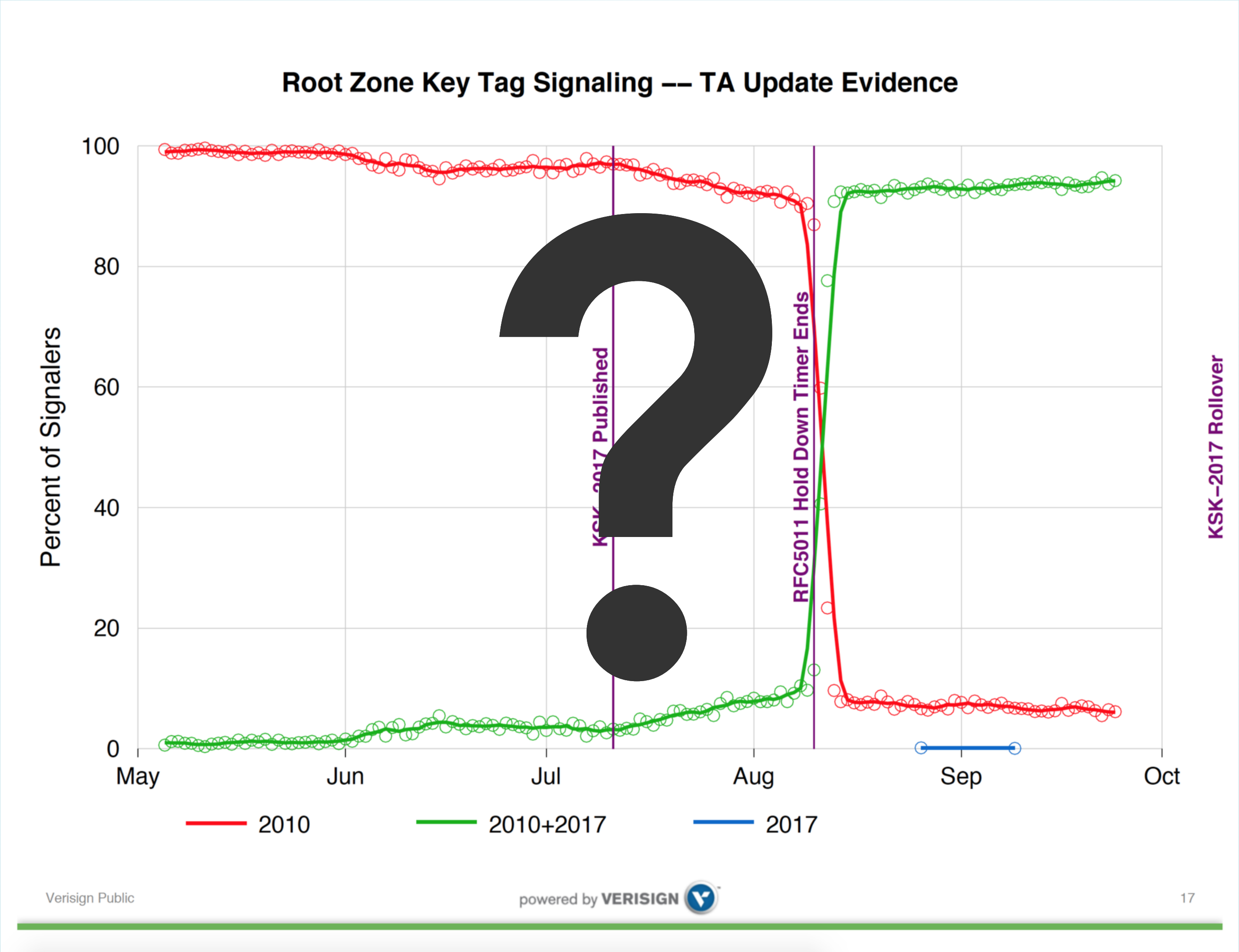
Abstract

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 **resolvers**

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

Verisign Public

powered by VERISIGN

17

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

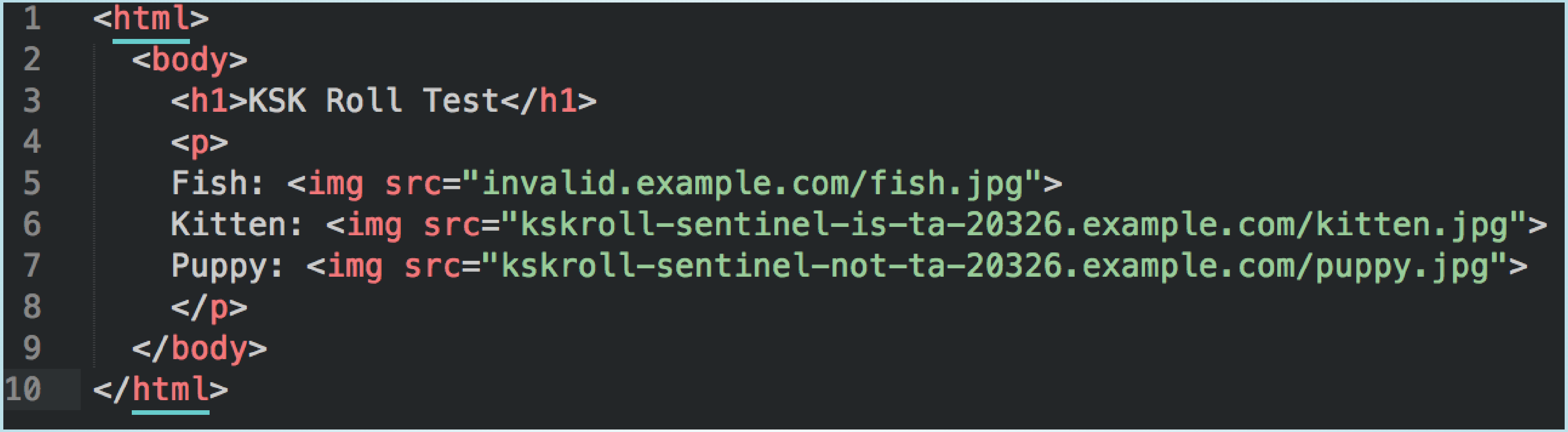
Just before sending the **response** (after resolution, validation):

- `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 resolve it and get 192.0.2.23
  - I have (and am using) KeyID 20326
    - answer with 192.0.2.23
- I get a query for `kskroll-sentinel-not-ta-20326.exam`
  - I do have (and am using) KeyID 20326
    - send SERVFAIL

Yawn. So what?!



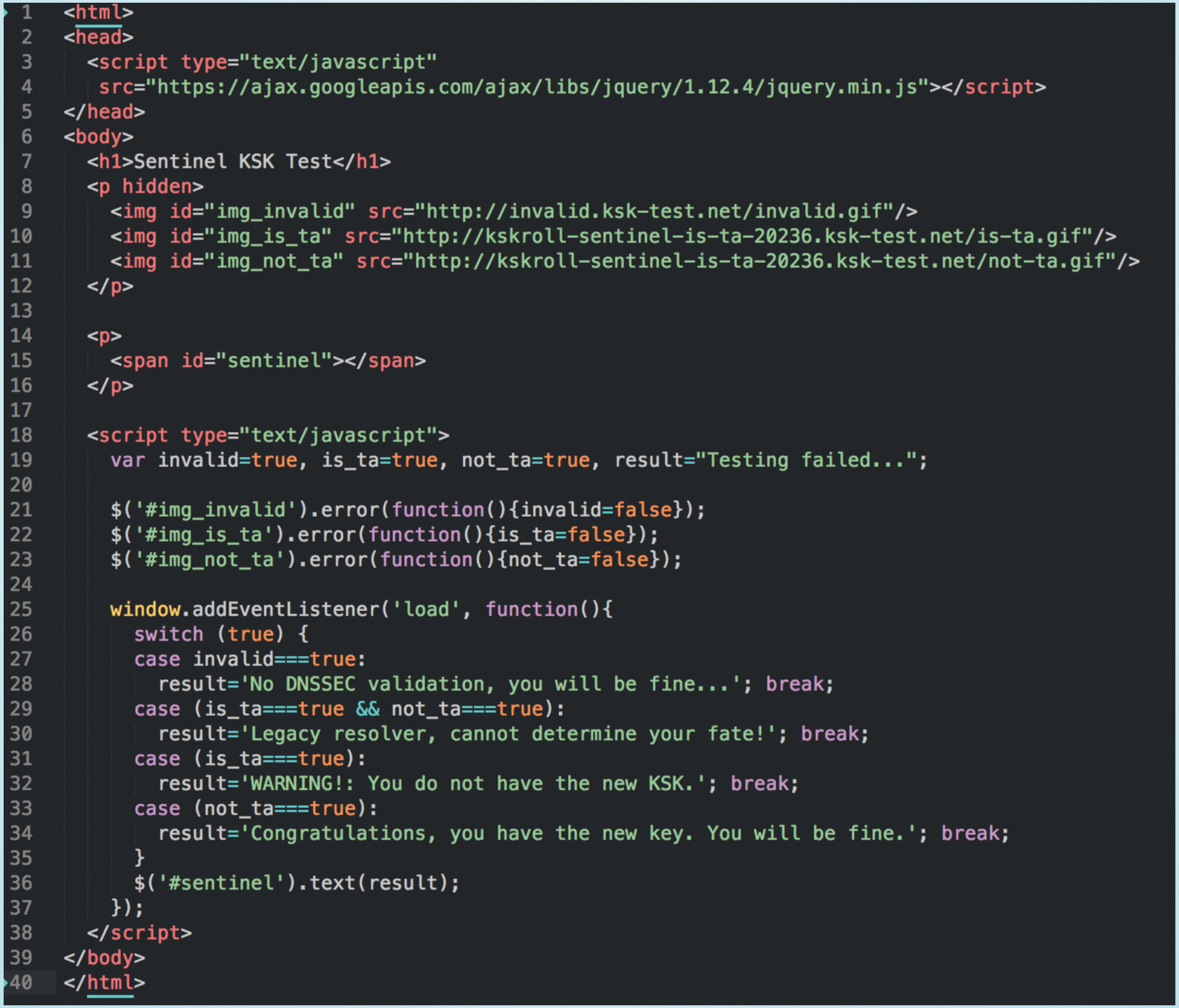
```
<html>
  <body>
    <h1>KSK Roll Test</h1>
    <p>
    Fish: <img src="invalid.example.com/fish.jpg">
    Kitten: <img src="kskroll-sentinel-is-ta-20326.example.com/kitten.jpg">
    Puppy: <img src="kskroll-sentinel-not-ta-20326.example.com/puppy.jpg">
    </p>
  </body>
</html>
```

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...

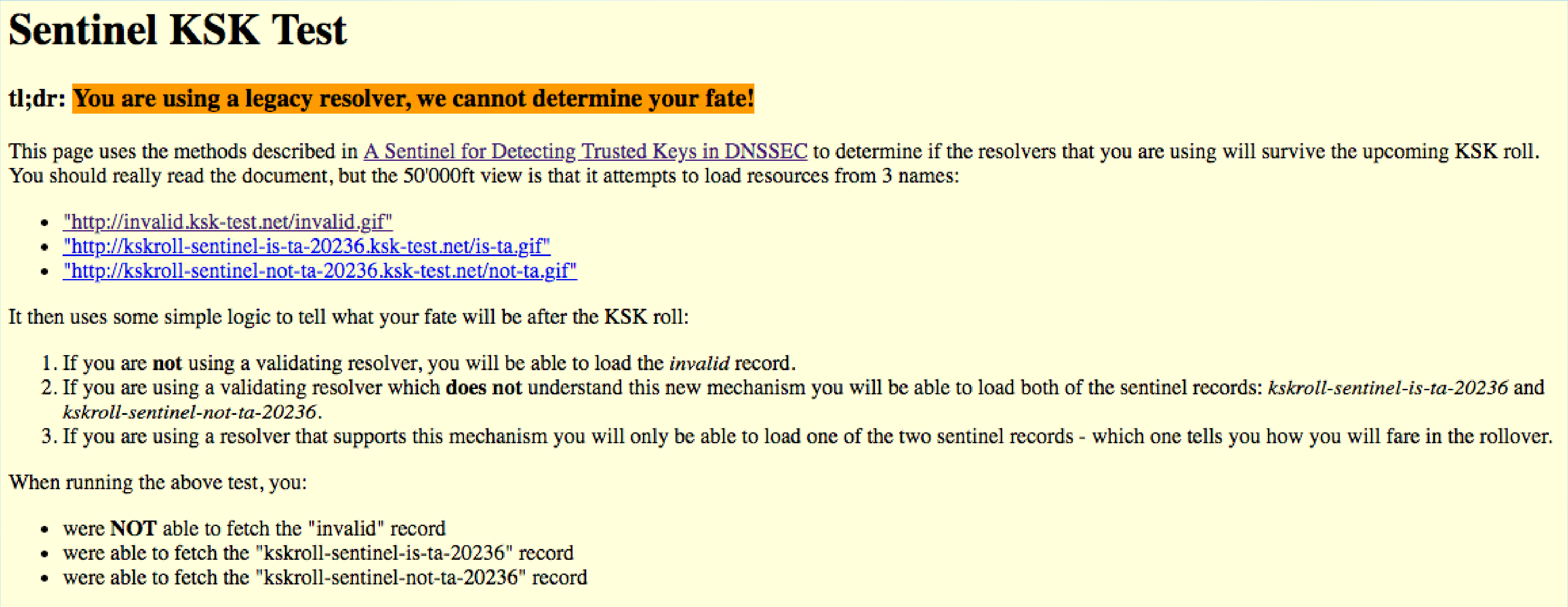


```
<html>
<head>
  <script type="text/javascript"
   src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
</head>
<body>
  <h1>Sentinel KSK Test</h1>
  <p hidden>
    <img id="img_invalid" src="http://invalid.ksk-test.net/invalid.gif"/>
    <img id="img_is_ta" src="http://kskroll-sentinel-is-ta-20236.ksk-test.net/is-ta.gif"/>
    <img id="img_not_ta" src="http://kskroll-sentinel-is-ta-20236.ksk-test.net/not-ta.gif"/>
  </p>

  <p>
    <span id="sentinel"></span>
  </p>

  <script type="text/javascript">
    var invalid=true, is_ta=true, not_ta=true, result="Testing failed...";

    $('#img_invalid').error(function(){invalid=false});
    $('#img_is_ta').error(function(){is_ta=false});
    $('#img_not_ta').error(function(){not_ta=false});

    window.addEventListener('load', function(){
      switch (true) {
      case invalid===true:
        result='No DNSSEC validation, you will be fine...'; break;
      case (is_ta===true && not_ta===true):
        result='Legacy resolver, cannot determine your fate!'; break;
      case (is_ta===true):
        result='WARNING!: You do not have the new KSK.'; break;
      case (not_ta===true):
        result='Congratulations, you have the new key. You will be fine.'; break;
      }
      $('#sentinel').text(result);
    });
  </script>
</body>
</html>
```

...but kittens!!!

Sorry, 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://kskroll-sentinel-is-ta-20236.ksk-test.net/is-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?