
DNSSEC failure
Now what?
A case study
.dk

I know this already, why are you telling me this?

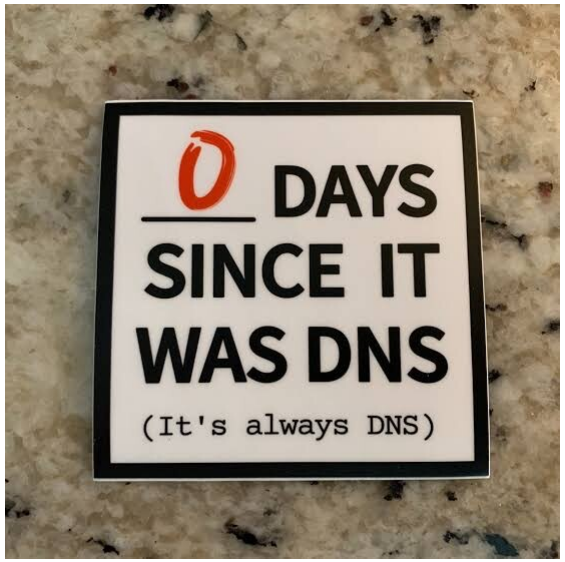
Yes, many of the next slides are known for many of you in ***this*** audience. Others may not.

But outside this room this is not widely known.

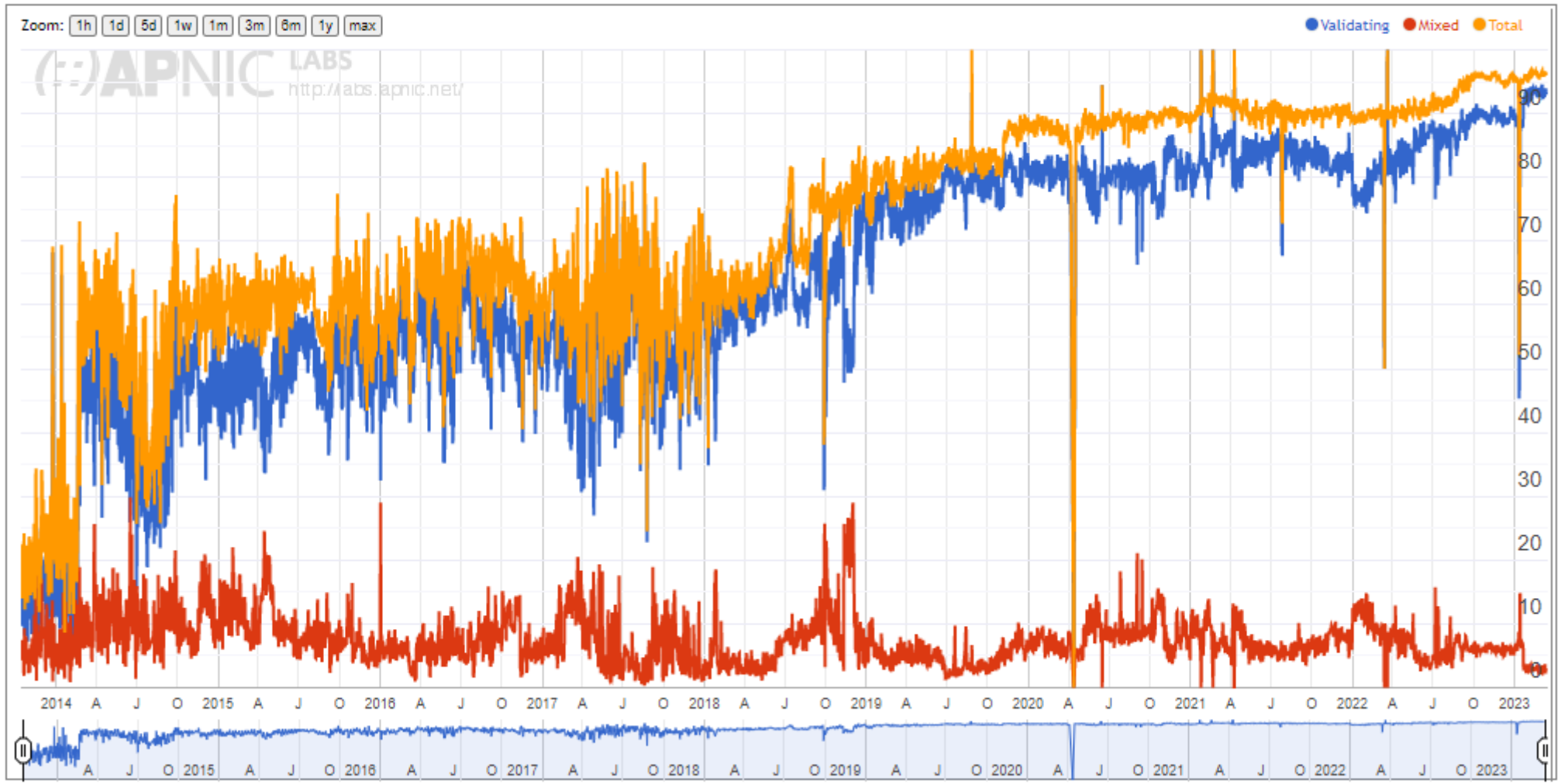
Go out and preach!

This presentation for the Danish national ISP ISAC came about after a case of DNS failures of a few nationally critical domains.

Spoiler alert: DNSSEC was not to blame.



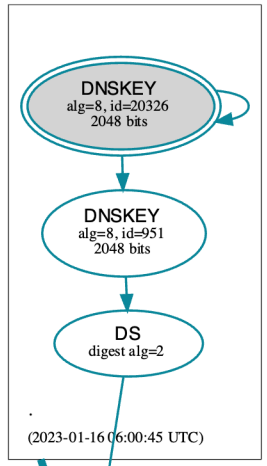
Use of DNSSEC Validation for Denmark (DK)



DNS

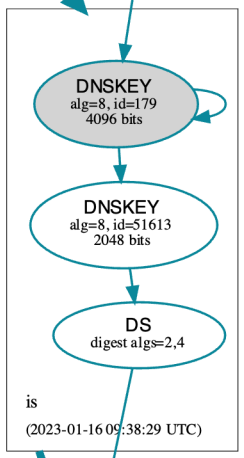
root

IANA (ICANN)



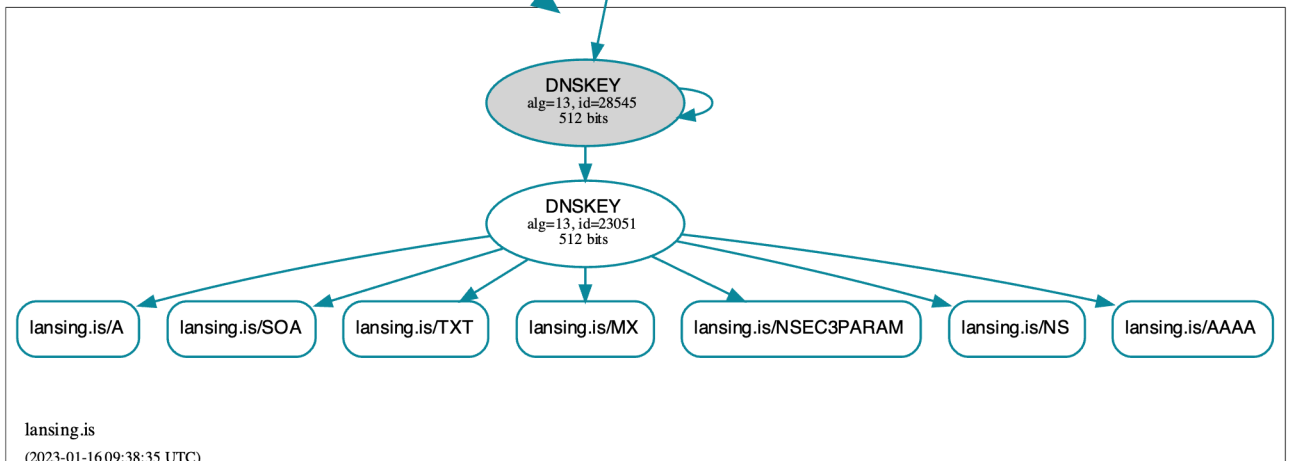
.is

TLD Registry
- ISNIC
- DK Hostmaster



lansing.is

DNS provider
- cloudflare
- one.com
- simply.com
- droso.dk



DNS

ISP
Recursive caching nameserver

End-user

root

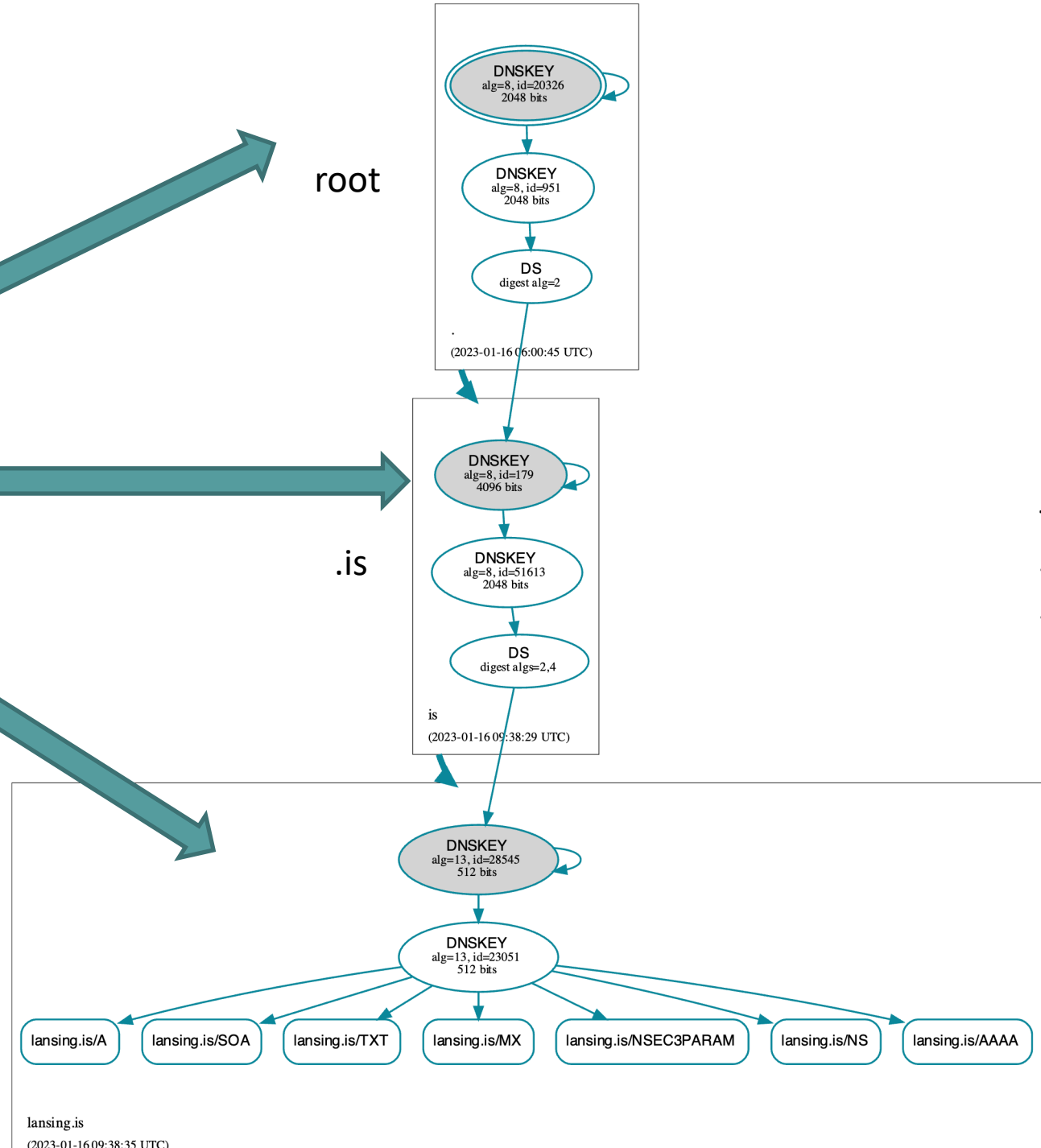
IANA (ICANN)

.is

TLD Registry
- ISNIC
- DK Hostmaster

lansing.is

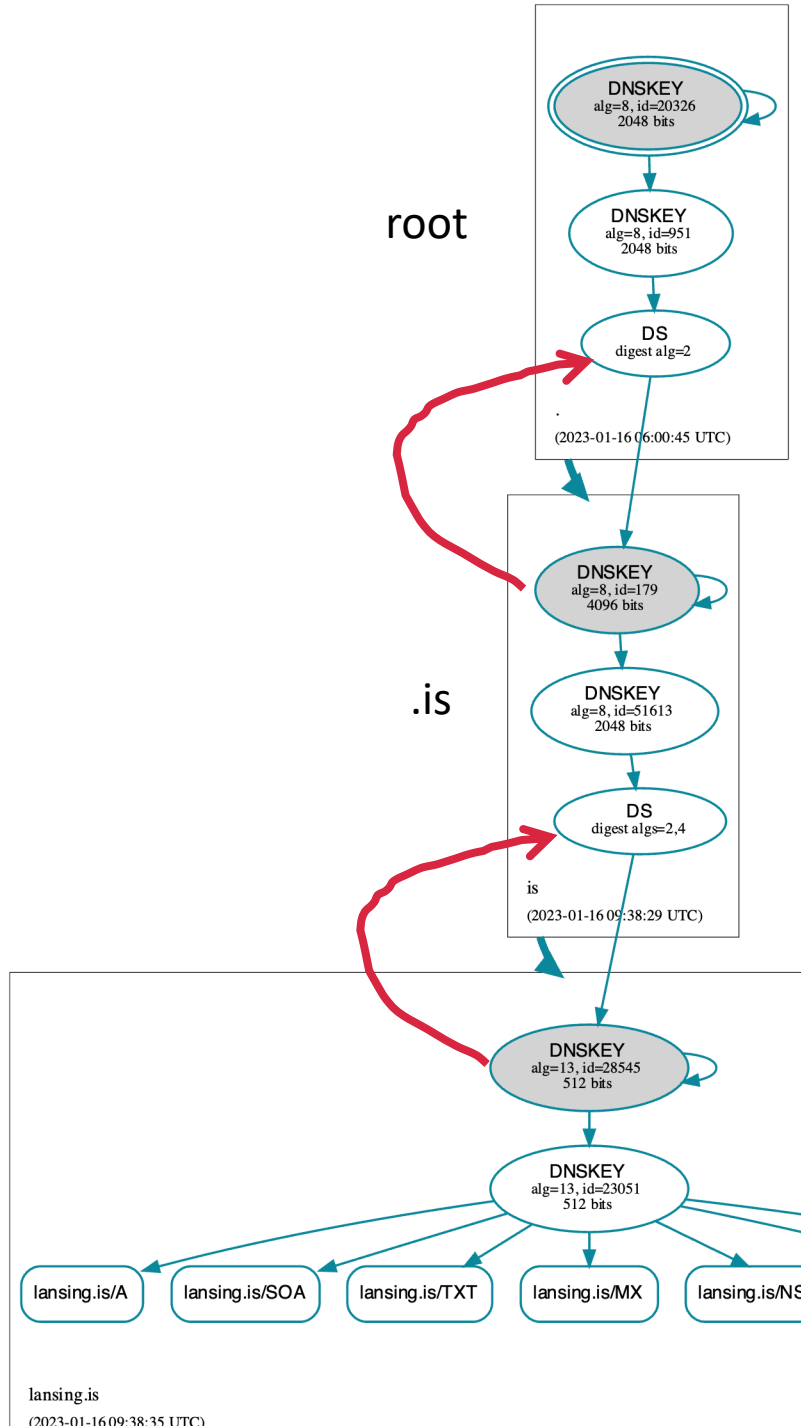
DNS provider
- cloudflare
- one.com
- simply.com
- droso.dk



DNSSEC

root

IANA (ICANN)



.is

TLD Registry
- ISNIC
- DK Hostmaster

lansing.is

DNS provider
- cloudflare
- one.com
- simply.com
- droso.dk

Why DNSSEC?

End-user/system security:

- Authenticity: The reply is coming from the requested domain name
- Integrity: The reply has not been modified in transit

No DNSSEC makes many attacks easier to perform:

- Man in the middle attack
- Cache poisoning
- Spoofing
- Simplified, generally – redirecting traffic the wrong way

How do you test changes to DNSSEC signing?

Depending on the change: a test zone or a full test platform

Algorithm rollover or *key rollover*

- Test procedure on a test zone* (or subdomain)
  - thiszoneisnotinuse.dk
  - thiszoneisnotinusee.erwin.dk
    - Does not test DS changes to the registry

Software upgrade

- Full test platform, incl. test zones

*: Or an unused or less used domain, like a defensive registration

Tools

dig(1)

+[no]adflag

Set [do not set] the AD (authentic data) bit in the query. This requests the server to return whether all of the answer and authority sections have all been validated as secure according to the security policy of the server. AD=1 indicates that all records have been validated as secure and the answer is not from a OPT-OUT range. AD=0 indicate that some part of the answer was insecure or not validated. This bit is set by default.

+[no]dnssec

Requests DNSSEC records be sent by setting the DNSSEC OK bit (DO) in the OPT record in the additional section of the query.

DNSViz | A DNS visualization tool

Zonemaster

SERVFAIL

What if something catastrophically fails for an import, *really* important, domain?

.dk

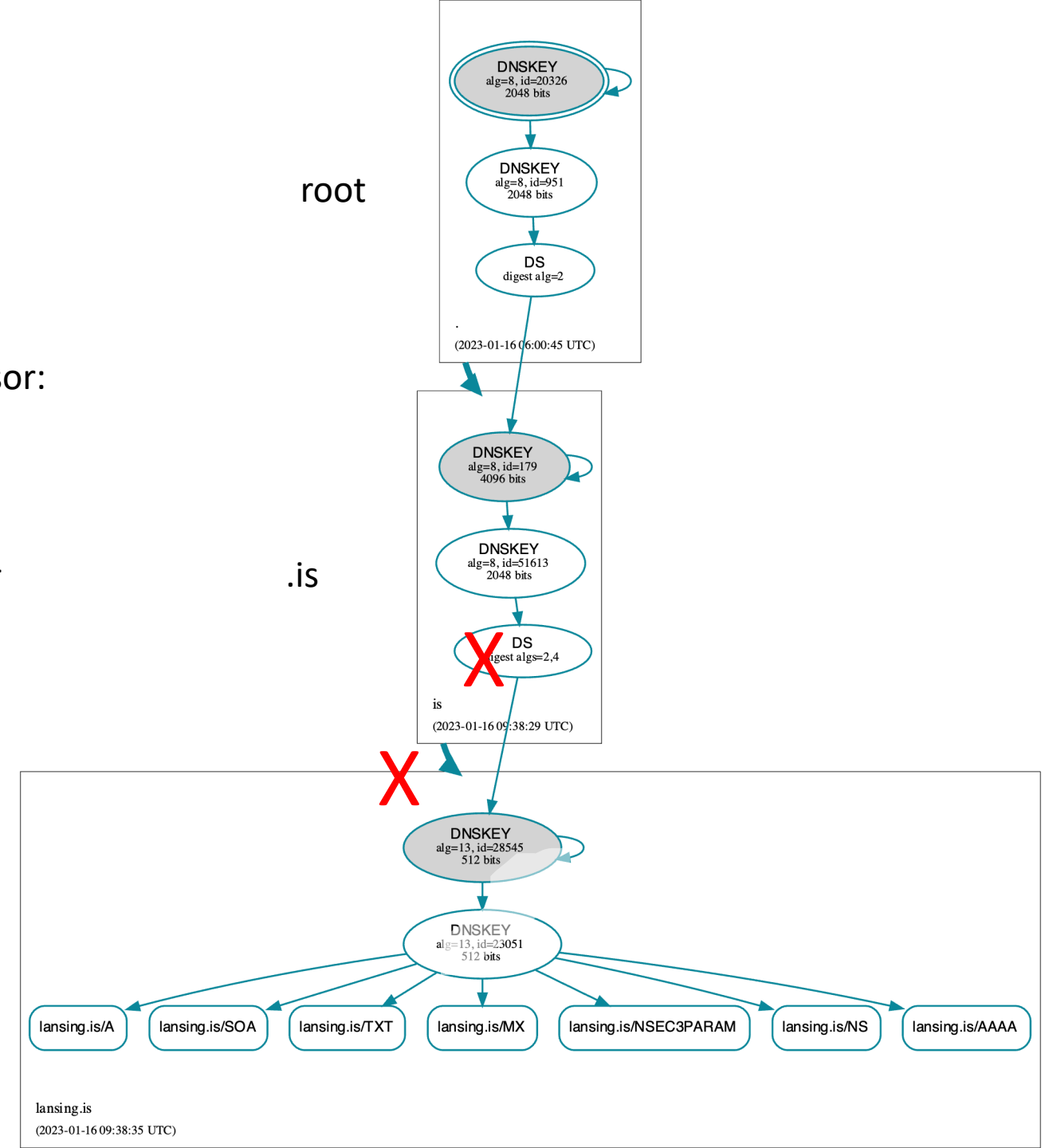
Dead man's switch 1

DS in parent signals to the recursor:

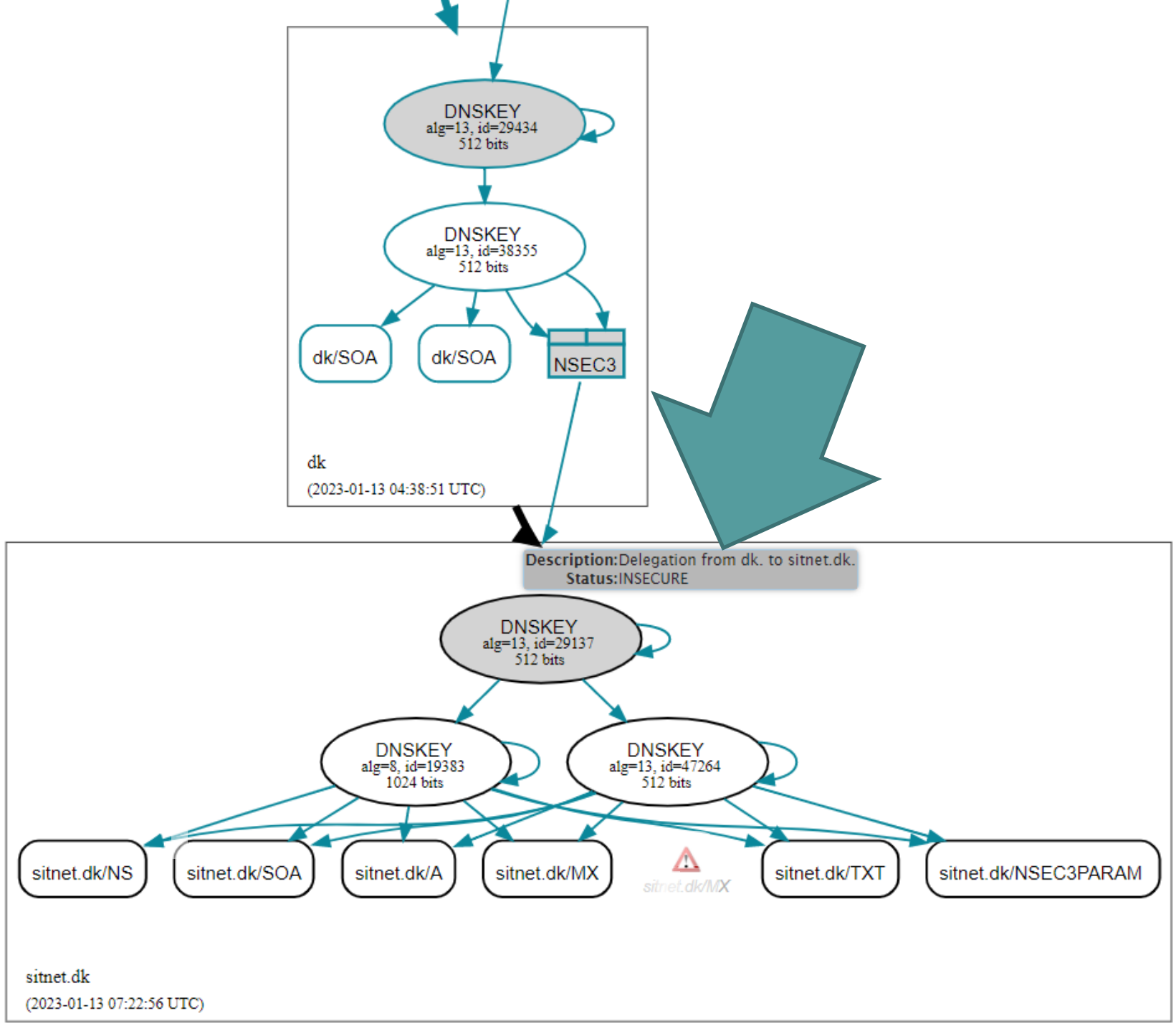
1. Zone **has to be** signed
   -> no signature: SERVFAIL
2. **Only** with this/these key(s)
   -> wrong signature: SERVFAIL

No DS in parent signals:

1. Zone is not signed
   - ➢ ignore all signatures



sitnet.dk


DNSKEY
alg=13, id=29434
512 bits
DNSKEY
alg=13, id=38355
512 bits
dk/SOA
dk/SOA
NSEC3
dk
(2023-01-13 04:38:51 UTC)
Description:Delegation from dk. to sitnet.dk.
Status:INSECURE
DNSKEY
alg=13, id=29137
512 bits
DNSKEY
alg=8, id=19383
1024 bits
DNSKEY
alg=13, id=47264
512 bits
sitnet.dk/NS
sitnet.dk/SOA
sitnet.dk/A
sitnet.dk/MX
sitnet.dk/MX
sitnet.dk/TXT
sitnet.dk/NSEC3PARAM
sitnet.dk
(2023-01-13 07:22:56 UTC)

Note about this case

From DNS operator: ”As it turned out, some components between some ISPs and us did not talk quite well to each other, so it did not have nothing to do with the algorithm rollover, as was the original suspicion.”

Not about this case

From DNS operator: ”As it turned out, some components between some ISPs and us did not talk quite well to each other, so it did not have nothing to do with the algorithm rollover, as was the original suspicion.”

Hint for network peoples on DNSSEC og IPv6:

IPv6 Fragmentation Loss | blabs (apnic.net)

We observed a V6 fragmentated packet drop rate of 21% in 2017.

What do we see in 2021?

The bottom-line answer is that current average drop rate for IPv6 Fragmented [illegible] considerable improvement over a relatively short period of time.

Code	Region	V6Frag Drop Rate	Drops	Sample Count
XA	World	6.14%	10,884,762	177,166,897
XC	Americas	11.09%	4,158,206	37,479,399
XE	Europe	9.03%	2,147,889	23,786,981
XB	Africa	6.93%	65,504	944,657
XG	Unclassified	5.99%	9,810	163,704
XF	Oceania	5.71%	40,559	710,371
XD	Asia	3.91%	4,462,794	114,081,785

Dead man's switch 1

- Remove DS in parent

.dk zone:

;; AUTHORITY SECTION:

```
erwin.dk.        86400  IN  NS  ns2.droso.dk.
erwin.dk.        86400  IN  NS  ns3b.droso.dk.
```

;; ANSWER SECTION:

```
erwin.dk.        7200   IN  DS  39526 13 2 A0D538EFEF6EFFA027DE2E19CF38FDFEF247A412937F208311DBB721 5BECC0BA
erwin.dk.        7200   IN  DS  39526 13 4 734FBE31FB86F9EFE179424A2BC5EF469A0FFF442D4D8BF300F020B0 31E15A04F416984750B7630E1BF2E370306A88C7
```

If you still can't wait:

Ask ISPs to flush the cache for this particular domain.

Small bang DNSSEC - Viktor Dukhovni (ICAN75): https://l.fnidder.dk/YP1nB
(do a whois lookup on fnidder.dk if you don't trust this URL shortener or ask ICANN for shorter links)

We’re ISPs? Can we do anything on our recursors?

And not just run away ☺

Dead man’s switch 2

Can one turn off DNSSEC validation for just ***one*** domain? You ask.

- Without turning off DNSSEC validation for *all domains* for *all customers*?

Dead man's switch 2

Can one turn off DNSSEC validation for just ***one*** domain? You ask.

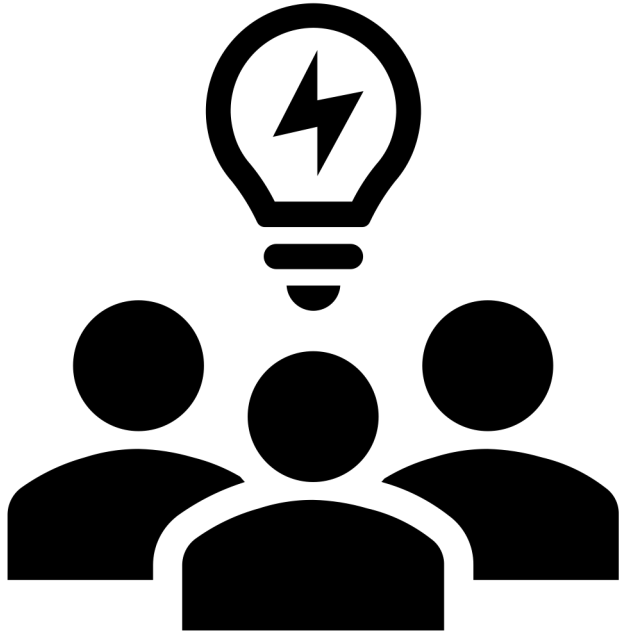
- Without turning off DNSSEC validation for *all domains* for *all customers*?

- Good of you to ask! You are not the first to ask that same question:

RFC 7646 Definition and Use of DNSSEC Negative Trust Anchors

Abstract

> DNS Security Extensions (DNSSEC) is now entering widespread deployment. However, domain signing tools and processes are not yet as mature and reliable as those for non-DNSSEC-related domain administration tools and processes. This document defines Negative Trust Anchors (NTAs), which can be used to mitigate DNSSEC validation failures by disabling DNSSEC validation at **specified domains**.



Dead man's switch 2

1. Introduction and Motivation

DNSSEC has now entered widespread deployment. However, the DNSSEC signing tools and processes are less mature and reliable than those for non-DNSSEC-related administration. As a result, operators of DNS recursive resolvers, such as Internet Service Providers (ISPs), occasionally observe domains incorrectly managing DNSSEC-related resource records. This mismanagement triggers DNSSEC validation failures and then causes large numbers of end users to be unable to reach a domain. **Many end users tend to interpret this as a failure of their ISP or resolver operator**, and they may switch to a non-validating resolver or **contact their ISP to complain**, rather than seeing this as a failure on the part of the domain they wanted to reach. Without the techniques in this document, this pressure may cause the resolver operator to disable (or simply not deploy) DNSSEC validation.

Dead man’s switch 2

Use of an NTA to **temporarily disable** DNSSEC validation for a specific misconfigured domain name immediately restores access for end users. This allows the domain's **administrators to fix their misconfiguration** while also allowing the organization using the NTA to keep **DNSSEC validation enabled and still reach the misconfigured domain**.

It is therefore

RECOMMENDED that NTA implementors should periodically attempt to validate the domain in question, for the period of time that the NTA is in place, until such validation is again successful. **NTAs MUST expire** automatically when their configured lifetime ends. The lifetime SHOULD NOT exceed a week.

Negative Trust Anchors: unbound

A.1. NLnet Labs Unbound

Unbound [Unbound-Config] lets us simply disable validation checking for a specific zone by adding configuration statements to unbound.conf:

```
server:
        domain-insecure: "foo.example.com"
```

Using the 'unbound-control' command, one can add and remove NTAs without restarting the name server.

Using the "unbound-control" command:

```
list_insecure               list domain-insecure zones
insecure_add zone           add domain-insecure zone
insecure_remove zone        remove domain-insecure zone
```

Items added with the "unbound-control" command are added to the running server and are lost when the server is restarted. Items from unbound.conf stay after restart.

Negative Trust Anchors: BIND

A.2. Internet System Consortium (ISC) BIND

Use the "rndc" command:

nta -dump
    List all negative trust anchors.
nta [-lifetime duration] [-force] domain [view]
    Set a negative trust anchor, disabling DNSSEC validation
    for the given domain.
    Using -lifetime specifies the duration of the NTA, up
    to one week. The default is one hour.
    Using -force prevents the NTA from expiring before its
    full lifetime, even if the domain can validate sooner.
nta -remove domain [view]
    Remove a negative trust anchor, re-enabling validation
    for the given domain.

Negative Trust Anchors: Nominum Vantio

A.3. Nominum Vantio

**

negative-trust-anchors

Format: name

Command Channel: view.update name=world negative-trust-anchors=(foo.example.com)

Command Channel: resolver.update name=res1 negative-trust-anchors=(foo.example.com)

Description: Disables DNSSEC validation for a domain, even if the domain is under an existing security root.

Negative Trust Anchors: knot-resolver

trust_anchors.set_insecure(*nta_set*)

Parameters

nta_list (*table*) – List of domain names (text format) representing NTAs.

When you use a domain name as an *negative trust anchor* (NTA), DNSSEC validation will be turned off at/below these names. Each function call replaces the previous NTA set. You can find the current active set in `trust_anchors.insecure` variable. If you want to disable DNSSEC validation completely use `trust_anchors.remove()` function instead.

Example output:

```
> trust_anchors.set_insecure({ 'bad.boy', 'example.com' })
> trust_anchors.insecure
[1] => bad.boy
[2] => example.com
```

What’s going on in .dk?

Teaser

RFC 9276 Guidance for NSEC3 Parameter Settings

2.3. Iterations

NSEC3 records are created by first hashing the input domain and then repeating that hashing using the same algorithm a number of times based on the iteration parameter in the NSEC3PARAM and NSEC3 records. The first hash with NSEC3 is typically sufficient to discourage zone enumeration performed by "zone walking" an unhashed NSEC chain.

Only determined parties with significant resources are likely to try and uncover hashed values, regardless of the number of additional iterations performed.

RFC 9276 Guidance for NSEC3 Parameter Settings

2.4. Salt

NSEC3 records provide an additional salt value, which can be combined with a Fully Qualified Domain Name (FQDN) to influence the resulting hash, but properties of this extra salt are complicated.

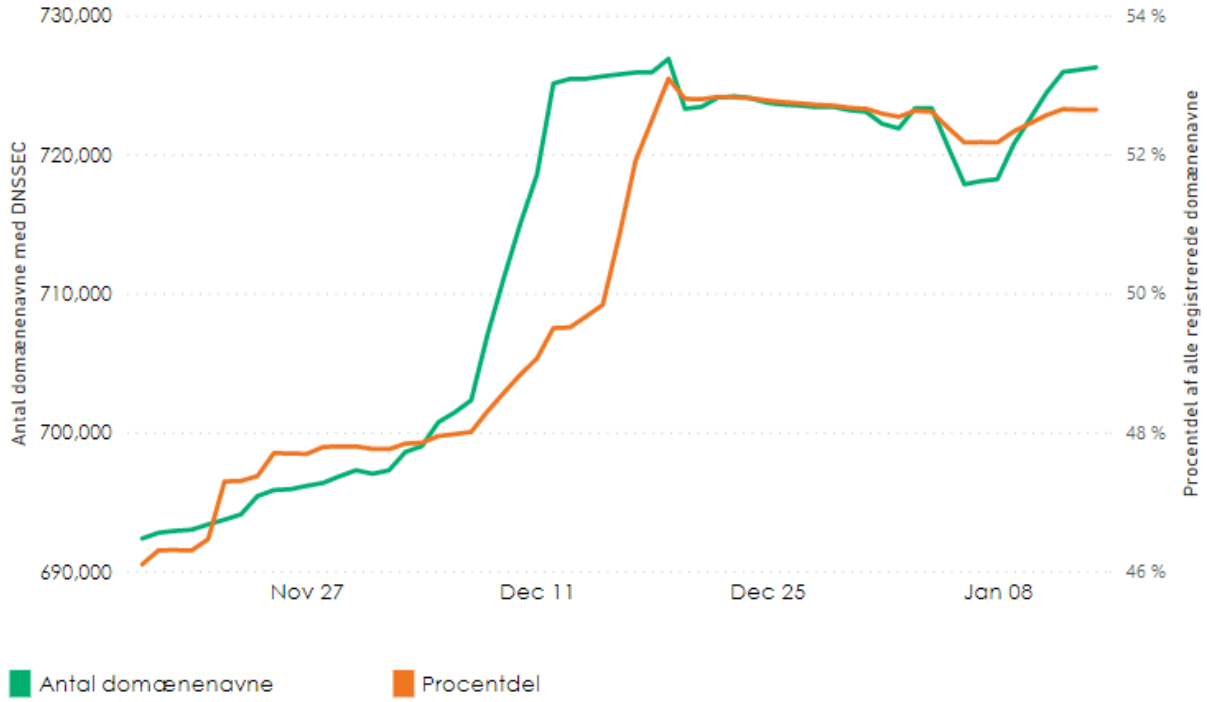
In the case of DNS, the situation is different because the hashed names placed in NSEC3 records are always implicitly "salted" by hashing the FQDN from each zone. Thus, no single pre-computed table works to speed up dictionary attacks against multiple target zones. An attacker is always required to compute a complete dictionary per zone, which is expensive in both storage and CPU time.

In other words, an additional, constant salt value does not change the cost of the attack.

RFC 9276 Guidance for NSEC3 Parameter Settings

Opt-out

For very large and sparsely signed zones, where the **majority** of the records are insecure delegations, opt-out MAY be used.



RFC 9276 Guidance for NSEC3 Parameter Settings

3.1. Best Practice for Zone Publishers

If NSEC3 must be used, then an iterations count of 0 MUST be used to alleviate computational burdens. Note that extra iteration counts other than 0 increase the impact of CPU-exhausting DoS attacks, and also increase the risk of interoperability problems.

For very large and sparsely signed zones, where the majority of the records are insecure delegations, opt-out MAY be used.

Operators SHOULD NOT use a salt by indicating a zero-length salt value instead (represented as a "-" in the presentation format).

RFC 9276 Guidance for NSEC3 Parameter Settings

In the near future, DK Hostmaster will change DNSSEC parameters for .dk:

- Reduce iterations to 0
- Remove salt
- Remove opt-out

To be continued…

Ørestads Boulevard 108, 11. sal
2300 København S

www.dk-hostmaster.dk