Authenticated Bootstrapping of DNSSEC Delegations

September 8, 2021

Peter Thomassen <peter@desec.io>

draft-thomassen-dnsop-dnssec-bootstrapping

DNSSEC validation rate

28 %

- 28% globally
- 50–95% in some places

vs.

secure delegation rate

5 %

- 5% globally
- 50–70% in some places
- **even for signed zones: < 50%**

Sources: deSEC, https://stats.labs.apnic.net/dnssec, https://rick.eng.br/dnssecstat/, https://www.sidn.nl/en/news-and-blogs/dnssec-adoption-heavily-dependent-on-incentives-and-active-promotion

But why?!

DNSSEC Bootstrapping Today ("How to Turn DNSSEC On")

– – –

- Securing delegation requires conveying DS/DNSKEY records to parent

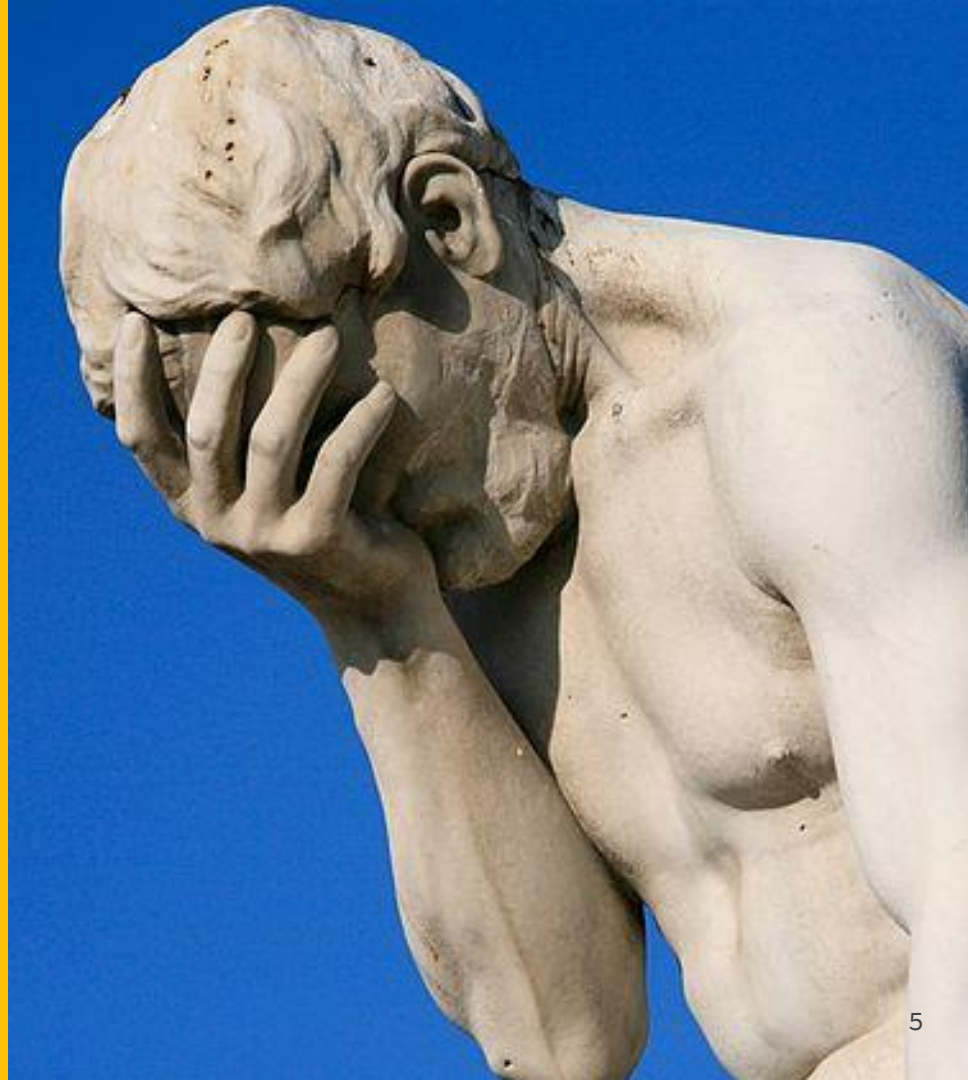
- Several approaches used by registrars / ccTLD registries:
  - trust on first use (TOFU, hope for the best)
  - manual submission by registrant/registrar (common and cumbersome)
  - REST interfaces (seems dead*)
  - CDS/CDNSKEY from insecure child (RFC 8078, requires stateful monitoring)

- Downsides: unauthenticated, out of band, slow, stateful, error-prone, too many parties, no automation / requires trigger, ...

* ICANN 54 (2015), draft-ietf-regext-dnsoperator-to-rrr-protocol (2018)

DNSSEC is too hard

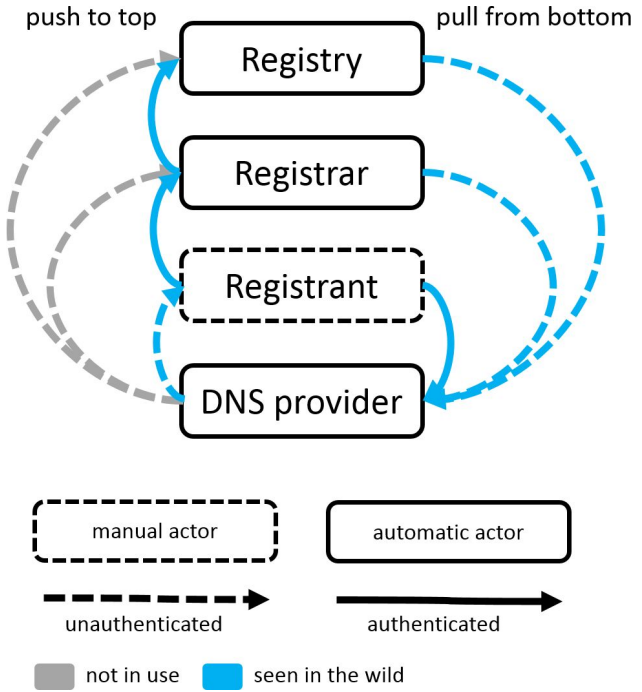
and we know it



Analysis: DS Signaling Model

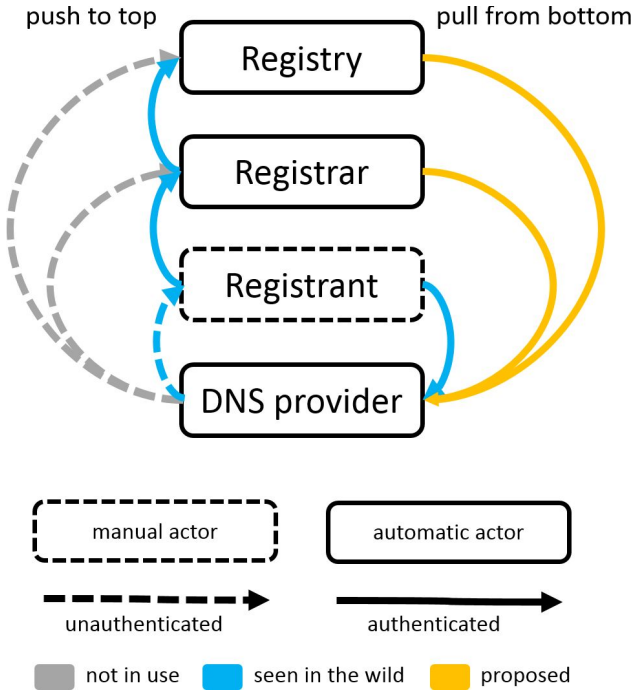
- Secure (authenticated) DS signaling currently involves many steps

- Reduce number of steps: make **registries / registrars pull directly from DNS provider**
  - so far not secure for DNSSEC bootstrapping



Analysis: DS Signaling Model

- Secure (authenticated) DS signaling currently involves many steps
- Reduce number of steps: make **registries / registrars pull directly from DNS provider**
  - so far not secure for DNSSEC bootstrapping
- **Goal:** authenticate pull from DNS provider



Solution Proposal:
Transferring Trust from the DNS Operator

Securing the `example.com` delegation (no existing DS)

Assumption: The **NS targets** (e.g. *`ns1.provider.net`*) **live in securely delegated zones** (e.g. *`provider.net`*).

(I) On the DNS provider side:

Publish `example.com`'s CDS/CDNSKEY records at a **"signaling name"** under the **nameserver zone**: `example.com.`*`ns1.provider.net`*

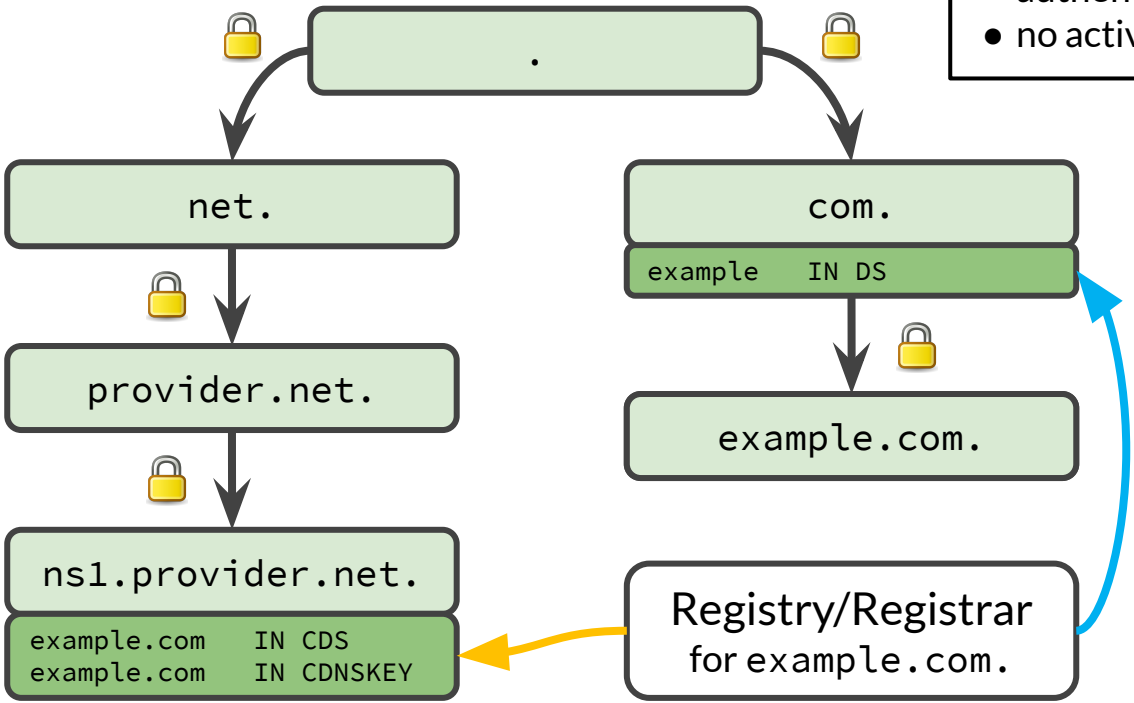
(II) On the registrar / ccTLD registry side:

When receiving a new NS record set,

1. **query** CDS/CDNSKEY records **from DNS provider** (using all NS names):
   - `example.com.`*`ns1.provider.net`*`,...`;
2. **validate**
   - **DNSSEC signatures** of responses,
   - **sanity check** (consistency with target zone);
3. **publish** `example.com`'s **DS records** in the parent zone → **done!** ✅

Trust Chain (example.com)

💡 Use an **established chain of trust** (left) to take a detour
- authenticated, immediate
- no active on-wire attacker



Technical Considerations

- No collision with other uses of CDS/CDNSKEY (those are apex-only)
- Zone names should be mapped onto an identifier (hash) to avoid hitting length constraints: **`h(example.com)`**`.ns1.provider.net`
- Add extra label to enable delegation of signaling data to separate zone: `h(...)`**`._boot`**`.ns1.provider.net`
- Advantages:
  - removes risk of accidentally modifying the nameserver's A/AAAA records
  - reduces churn on nameserver zone
  - allows zone to be operated on a different set of servers

Recap: We got ...

Signaling

- of **zone-specific** information
- from the **NS operator**
- **to the public** (e.g. the parent)

... which is

- **authenticated,**
- **in-band,**
- **immediate,**
- requires **no third parties**.

What else can be done with it?

Multisigner Key Exchange (in a Nutshell)

– – –

Multisigner Goals (RFC 8901):

- **Redundancy:** multi-homed zones with full validation of responses
- **Integrity:** smooth transition during provider transfer **(no going insecure)**

How it works:

- Operators **advertise each others' ZSKs** via the DNSKEY set that they sign;
- Parent **advertises all of the KSKs via its DS records**.

How can operators learn each other's ZSKs?

- Publish them in a DNSKEY RRset at `example.com.`*`ns1.other.net`*
- **Same signaling mechanism** as for DS bootstrapping

Phew. — How about some numbers?

What's needed for deployment?

- Secure signaling **requires that NS targets** are **in securely delegated zones**
  - if already the case: simplifies deployment for DNS operators
  - if not: overhead for DNS operator seems manageable

- DS bootstrapping **requires that NS targets** are **not part of the same zone**
  - **mostly the case:** > 99% of NS targets are out of bailiwick
    in bailiwick: < 0.33% for .com, < 0.72% for .net (thanks to John R. Levine)

- … and obviously, the zone itself needs to be signed.

- Survey time!

Survey on Deployment Requirements

– – –

- Analyze **top 1M sites** (Tranco dataset)

- For each domain in the dataset, extract
  a. whether the domain itself is **secure** (has validation path),
  b. whether there zone itself is **signed** (has RRSIGs),
  c. **all NS targets** in the delegation,
  d. which NS targets are **secure** (if any),

  … and compute things like
  Bootstrappability: What fraction of domains have a == false, but c == d?

- Measurements done by Nils Wisiol (huge thanks!)

deSEC

Survey on Deployment Requirements: General Results

```
Failure rate ..................................:   3.80%
Remaining sample size .........................:  962012

Proportion of secure zones ....................:   4.47%
Proportion of signed zones ....................:   5.87%

Proportion of zones with all nameserver targets secure:  24.14%
Proportion of zones with ≥ 1 nameserver targets secure:  25.36%
```

bootstrappable:
domain is not secure *and* NS targets have validation path → signaling possible

```
Proportion of bootstrappable zones (all NS) ...:  21.77%
Proportion of bootstrappable zones (≥ 1 NS) ...:  22.66%
```

deSEC

Survey on Deployment Requirements: by TLD and Operator

tld	domain count	bootstrapable mean	bootstrapable sum
com	493152	0.235917	116343
org	68720	0.180384	12396
net	43894	0.236274	10371
ru	31435	0.137649	4327
uk	20102	0.188936	3798
in	9208	0.287250	2645
io	7134	0.343706	2452
co	7089	0.302723	2146
de	27158	0.072833	1978
au	7964	0.242843	1934

ns_rname	bootstrapable	domain count
dns.cloudflare.com.	True	188746
dns.hostinger.com.	True	3436
hostmaster.nsone.net.	True	2470
NaN	True	1959
hostmaster.cscdns.net.	True	1393
postmaster.iij.ad.jp.	True	927
root.v1.wpxhosting.com.	True	639
nsadmin.nic.in.	True	563
dns.ds.network.	True	530
hostmaster.infomaniak.ch.	True	454

Thank you!

… also to our sponsors:

Questions?

Backup

Open Questions (I)

- If a DNS operator deploys DS bootstrapping, parents may like bulk processing. How is that best achieved?
  - allow NSEC walking of signaling zone (thanks to Brian Dickson)
  - allow public AXFR of signaling zone (thanks to John R. Levine)
  - both require a PTR record to map the signaling name onto the zone name
- Should an extra layer be inserted in `h(...)._boot.ns1.provider.net` to allow parent-specific bulk processing? (thanks to John R. Levine)
  - `h(example.co.uk).h(co.uk)._boot.ns1.provider.net`
  - compatible with both NSEC walking
  - also compatible with AXFR (but benefit gained only when using subzones for large parents)
- When NS RRset is received at registration, zone may not yet be operational
  - What else would be a good bootstrapping trigger for the registry/registrar?

Open Questions (II)

- Drop requirement that CDS/CDNSKEY within the target zone must match?
  - Having this gives all RFC 8078 guarantees incl. **opt out** (plus, a sanity check)
- Drop requirement that all NS responses must agree?
  - In particular, is it justifiable that only one auth has the bootstrapping records?
    - … but require that all must serve the usual records from the child zone itself?
  - May reduce deployment complexity / help protocol adoption
- Registries/registrars can select which TLDs to trust in the chain. Desirable?
  - One could say that you can't trust a DNS operator anyway if its NS hostnames are not trusted
  - On the registered NS domain (not TLD), trust problems can also occur if the DNS operator doesn't own the domain, or uses an external managed service
    - Ideally, DNS operator a) owns the zone, and b) either operates or presigns it

Security Model

- We use an established chain of trust to take a detour
  - authenticated, immediate
  - no active on-wire attacker
- Actors in the chain of trust can undermine the protocol
  - can also undermine CDS / CDNSKEY from insecure
  - but: known point in time / window of opportunity much smaller
- Further mitigations exist, e.g:
  - monitor delegation
  - diversify NS TLDs
  - multiple vantage points

deSEC

	Bootstrapping Method		
	Manual	CDS/CDNSKEY	Proposed
Bootstrapping Involves			
zone operator Z	✓[1]	✓	✓
domain owner	✓	✗	✗
registrar	✓	✗	✗
registry	✓	✓	✓
Actors who can initialize keys			
Required parties (trusted)			
registrar	✓	✓[2]	✓[2]
NS zone operator	✗	(✓)	(✓)[3]
NS zone ancestors	✗	(✓)	(✓)
NS zone owner	✗	(✓)	(✓)
Others parties (untrusted)			
active on-wire attacker	depends	✓[4]	✗
social engineering attacker [1]	✓	✗	✗
Properties			
Prerequisites	out-of-band channel	MITM attack mitigation	suitable NS zone configuration
Authentication	bad in practice [1]	none	cryptographically
Duration	varies	days	minutes

Table 1: Comparison of methods for establishing a new secure delegation, dispaying a) entities involved in the bootstrapping of an individual insecure zone, b) attack surface towards trusted and untrusted third parties, and c) prerequisites, key material authentication, and bootstrapping duration. Key initialization within parentheses (✓) requires collusion across all NS zones. [1] For offline signing, only the signing key holder is involved. [2] Registry could refuse deployment through registrar. [3] Requires knowledge of private key. [4] Several vantage points and long time must be covered.