Automatic DNSSEC Bootstrapping

using Authenticated Signals from the Zone's Operator

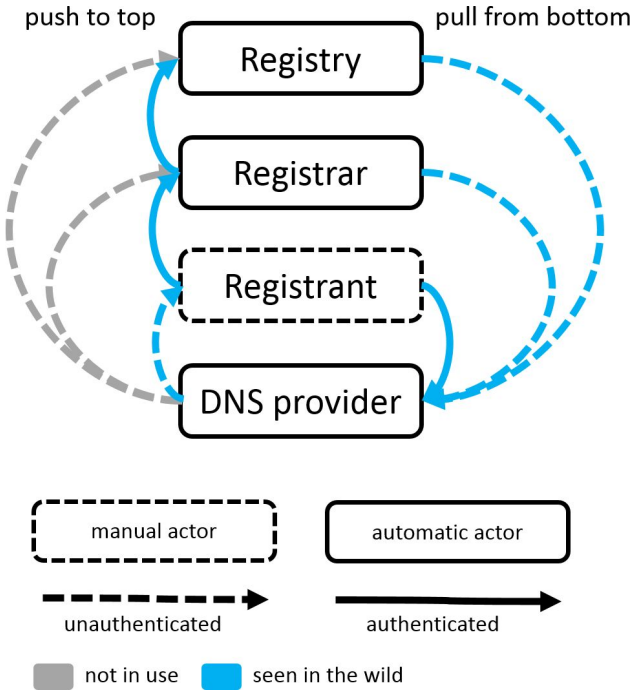
draft-thomassen-dnsop-dnssec-bootstrapping

IETF 112 – DNSOP WG
November 12, 2021

Peter Thomassen (deSEC, Secure Systems Engineering)
Nils Wisiol (deSEC, Technische Universität Berlin)

DS Bootstrapping and Why It Needs Improvement

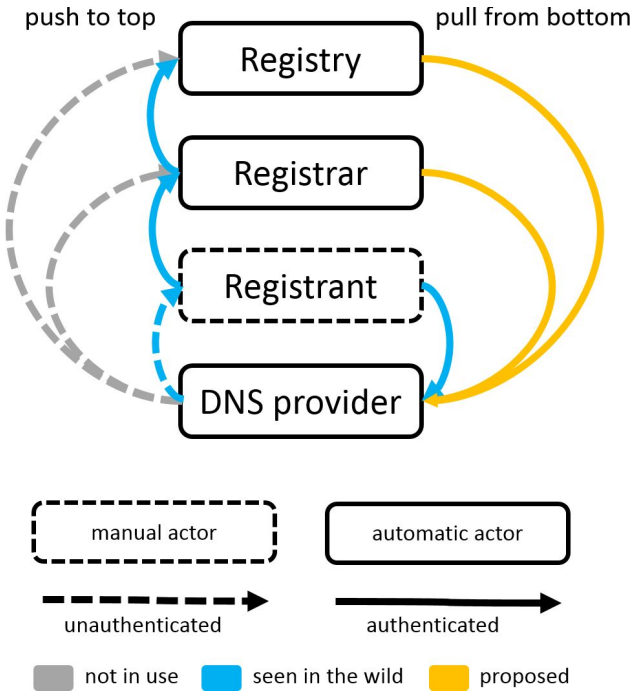
- Various methods have emerged
  - TOFU, manual submission, REST interfaces*, CDS/CDNSKEY from insecure (RFC 8078)
- Each suffers from one or more downsides
  - unauthenticated || out of band || slow || stateful || error-prone || too many parties || no automation
  - **Authenticated workflow involves too many steps**
- Promising: **direct pull from DNS operator**
  - RFC 8078 specifies this **in-band** (via CDS / CDNSKEY), but not secure for bootstrapping



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

DS Bootstrapping and Why It Needs Improvement

- Various methods have emerged
  - TOFU, manual submission, REST interfaces*, CDS/CDNSKEY from insecure (RFC 8078)
- Each suffers from one or more downsides
  - unauthenticated || out of band || slow || stateful || error-prone || too many parties || no automation
  - **Authenticated workflow involves too many steps**
- Promising: **direct pull from DNS operator**
  - RFC 8078 specifies this **in-band** (via CDS / CDNSKEY), but not secure for bootstrapping
  - **Goal: add authentication**
    → automatable, immediate, no state required



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

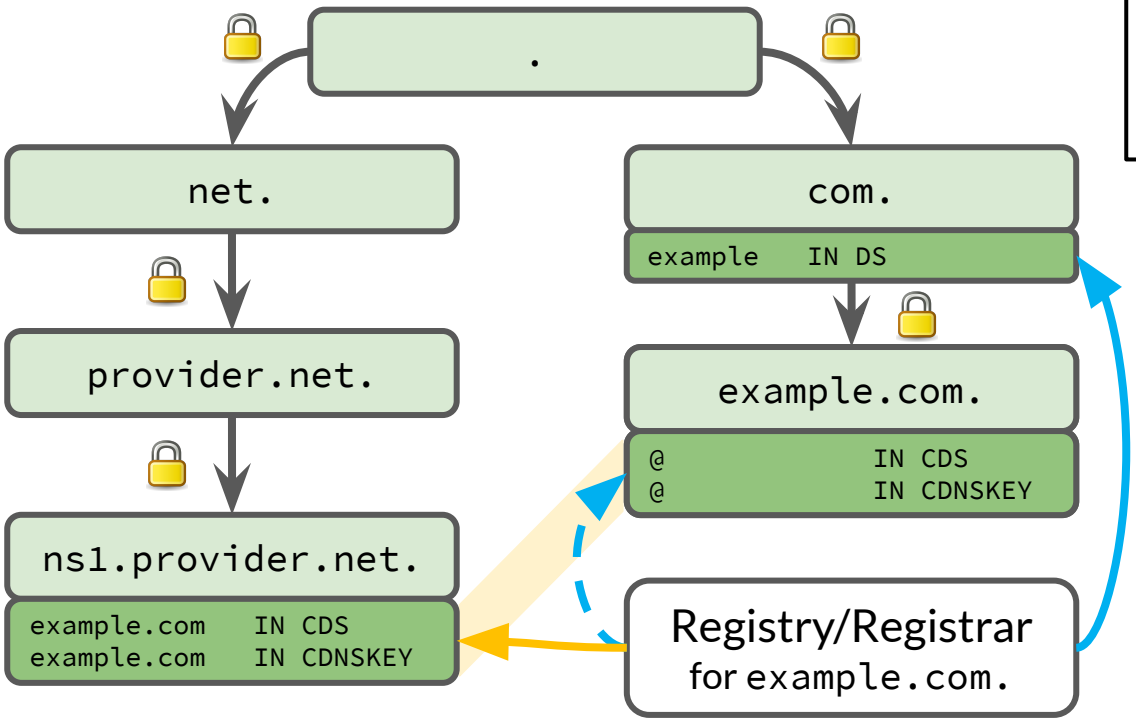
Proposed Solution: Transfer Trust from the DNS Operator

1. Create a **signaling mechanism for DNS operators**
   - **What?**
     - allow **publishing arbitrary information** about the zones they are authoritative for
     - in an **authenticated** fashion, **on a per-zone basis**
   - **How?**
     - use namespace **under each nameserver hostname**, e.g. _boot.ns1.desec.io
     - **require DNSSEC** under this namespace (requires nameserver domains to be secure)
     - under this namespace, **announcements** are made **using zone-specific owner names**

2. Use this mechanism to **publish an authentication signal**
   - start with **CDS/CDNSKEY records at the apex** of the target zone (RFC 8078)
   - **co-publish these records using the signaling mechanism** (signed with NS zone's keys)

3. **Validate** the target domain's CDS/CDNSKEY records **against this signal**
   - if successful: "transfer trust to the target domain" → **provision DS records** at the parent
   - **clean up** records when done

CDS Authentication: Co-Publish under Trusted Hostname



💡 Use an **established chain of trust** (left) to take a detour

- authenticated, immediate
- no active on-wire attacker

Technical Considerations

- No collision with primary use of CDS/CDNSKEY (those are apex-only)
- Replace ancestor labels with hash: **example.h(com)**.ns1.provider.net
  - to avoid hitting length constraints, and to allow per-parent handling
  - up for discussion (later)
- Add extra label: example.h(…).**_boot**.ns1.provider.net
  - to enable delegation of signaling data to separate zone
  - precise naming tbd (suggestion: _dsbootstrap)
- Name scheme features:
  - removes risk of accidentally modifying the nameserver's A/AAAA records
  - reduces churn on nameserver zone
  - allows splitting off DNS operations (e.g. online-signing with different key; delegate by parent)
  - allows parent to discover bootstrappable domains under h(parent)._boot (XFR, NSEC walk)

Numbers, numbers, numbers ...!

Survey on Deployment Requirements

- 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 Levine)

- Secure signaling **requires NS targets** to be **in securely delegated zones**
  - How frequent is that?
  - For each domain in **Tranco Top 1M dataset**, extract
    a. whether the domain itself is **secure** (has validation path),
    b. **all NS targets** in the delegation,
    c. which NS targets are **secure** (if any),

    … and compute things like
    Bootstrappability: A domain is *bootstrappable* if b == c, but a == false

Survey on Deployment Requirements: Bootstrappability

```
Measurement failure rate.............................:   2.30%
Remaining sample size................................:  977007

Proportion of secure zones...........................:   5.43%
Proportion of signed zones...........................:   6.84%

Proportion of zones with all nameserver targets secure:  24.63%
Proportion of zones with ≥ 1 nameserver targets secure:  25.97%
```

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

```
Proportion of bootstrappable zones (all NS) ..........:  22.11%
Proportion of bootstrappable zones (≥ 1 NS) ..........:  23.07%
```

as of 22 October 2021

Survey on Deployment Requirements: by TLD, by Provider

	zones	signed	secure	bootstrappable	
	total count	rel.	rel.	rel.	abs.
tld					
com	513660	4.5%	3.4%	23.2%	119195
org	71332	4.8%	3.7%	17.8%	12664
net	46232	6.8%	5.4%	22.1%	10231
ru	32387	7.3%	2.0%	13.9%	4511
uk	21003	4.3%	3.4%	18.8%	3945
in	9595	7.3%	5.7%	28.3%	2719
io	7673	8.6%	6.2%	34.9%	2677
xyz	4054	6.1%	5.1%	55.6%	2254
co	7408	10.6%	8.7%	29.7%	2201
online	3202	3.3%	2.4%	68.1%	2180

	zones	signed	secure	bootstrappable	
	total count	rel.	rel.	rel.	abs.
ns_rname					
dns.cloudflare.com.	252145	6.1%	3.1%	76.5%	192895
dns.hostinger.com.	4141	0.1%	0.0%	87.8%	3634
hostmaster.nsone.net.	19911	1.1%	0.9%	12.9%	2568
nan	80403	9.2%	8.6%	2.6%	2066
hostmaster.cscdns.net.	6041	1.8%	1.7%	22.8%	1375
dns.openprovider.eu.	1290	1.0%	0.8%	91.7%	1183
postmaster.iij.ad.jp.	935	2.0%	2.0%	98.0%	916
nstld.verisign-grs.com.	8531	90.4%	90.4%	7.5%	637
root.v1.wpxhosting.com.	617	0.3%	0.3%	99.7%	615
nsadmin.nic.in.	771	29.4%	29.4%	70.6%	544

as of 22 October 2021, “nan” ns_rname means that referenced NS zones have more than one rname in their SOAs

Discussion Point: **Do we want the hashed label?**

Do the benefits justify the added complexity?

Pros: … yes, please, hash please!

- Helps **stay within limits**
  - length / no. of labels → less edge cases
- **Prevents CDS ambiguity** at zone cut
  - What does foo.bar.net._boot.[…] mean?
  - It's possible that bar.net is not delegated
- **Improves privacy** during discovery
  - must know ancestor to begin NSEC walk
- **Flat structure**
  - simplifies scanning logic
  - facilitates adding prefixes → **"properties"** … like: _cds.**example.h(co.uk)**._signal.[…]

Cons: … no, smash the hash!

- **Complicates implementation**
  - all tooling needs to be able to hash
- Makes **debugging more difficult**
  - standard tools should suffice (dig etc.)
- Makes **synthesis more difficult**
  - How to dynamically associate an incoming query with a target domain? → **mapping needed (ancestors only!)**
  - h(co.uk)._boot DNAME co.uk._boot (cacheable per parent!)

What now?

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**

- Proposing to use this channel for authenticating CDS/CDNSKEY records
  - Some parties have expressed interest, and potential seems high
  - Perhaps other uses will emerge in the future
- Need to settle on a naming scheme (“to hash or not to hash”)
- Would the WG be interested in the adopting this draft?