DNSSEC Bootstrapping Support in Knot DNS 3.3.5 and in PowerDNS

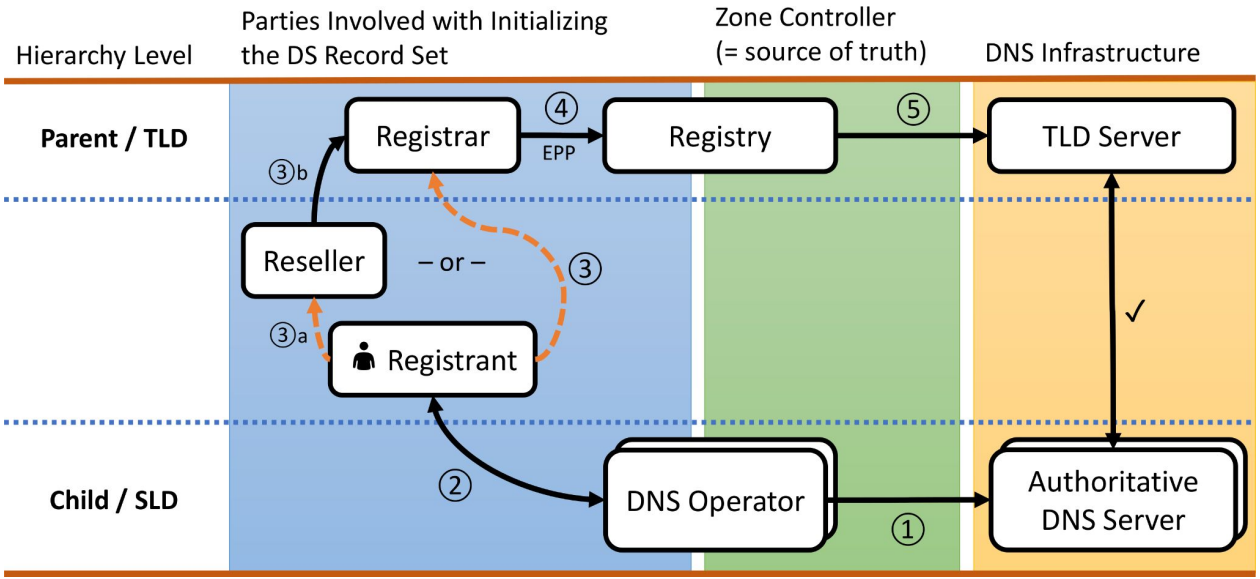
draft-ietf-dnsop-dnssec-bootstrapping

Peter Thomassen <peter@desec.io>

RIPE 88 – DNS Working Group
May 22, 2024

deSEC

Registrant-centric DS Provisioning

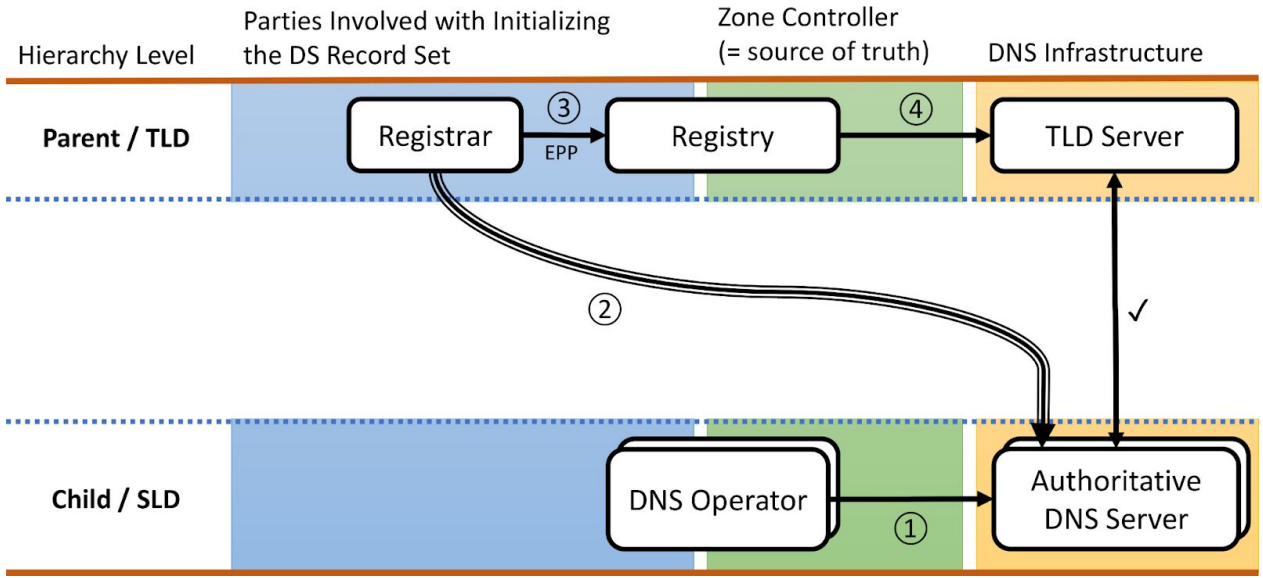


- Slow
- Error-prone
- Out of band
- Authenticated (in theory)

- Involves the Child DNS Operator (origin) and Parent Registry (recipient)
  - … typically with the Registrar as the messenger
  - … typically facilitated through the Registrant

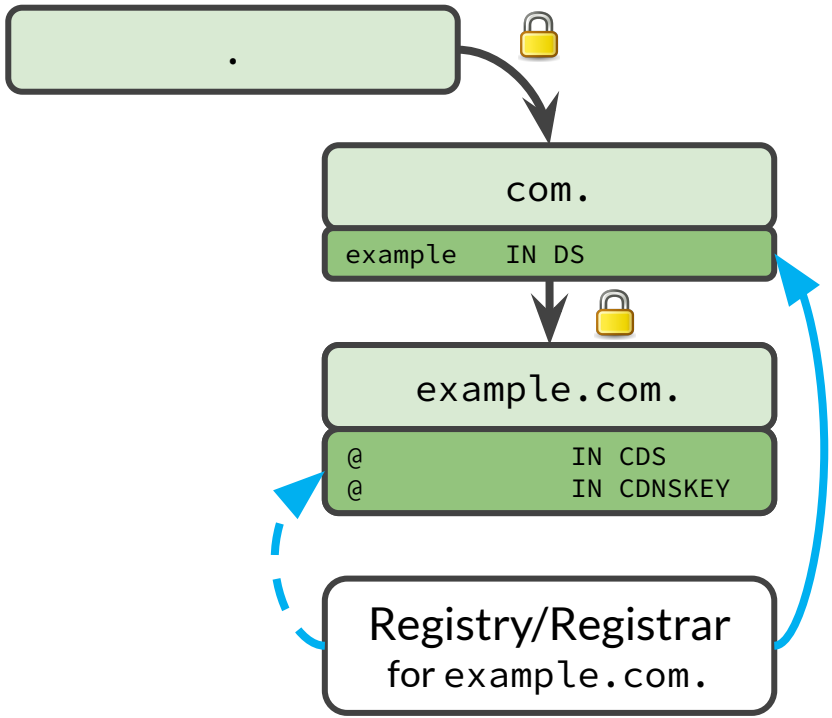
deSEC

Automatic DS Provisioning



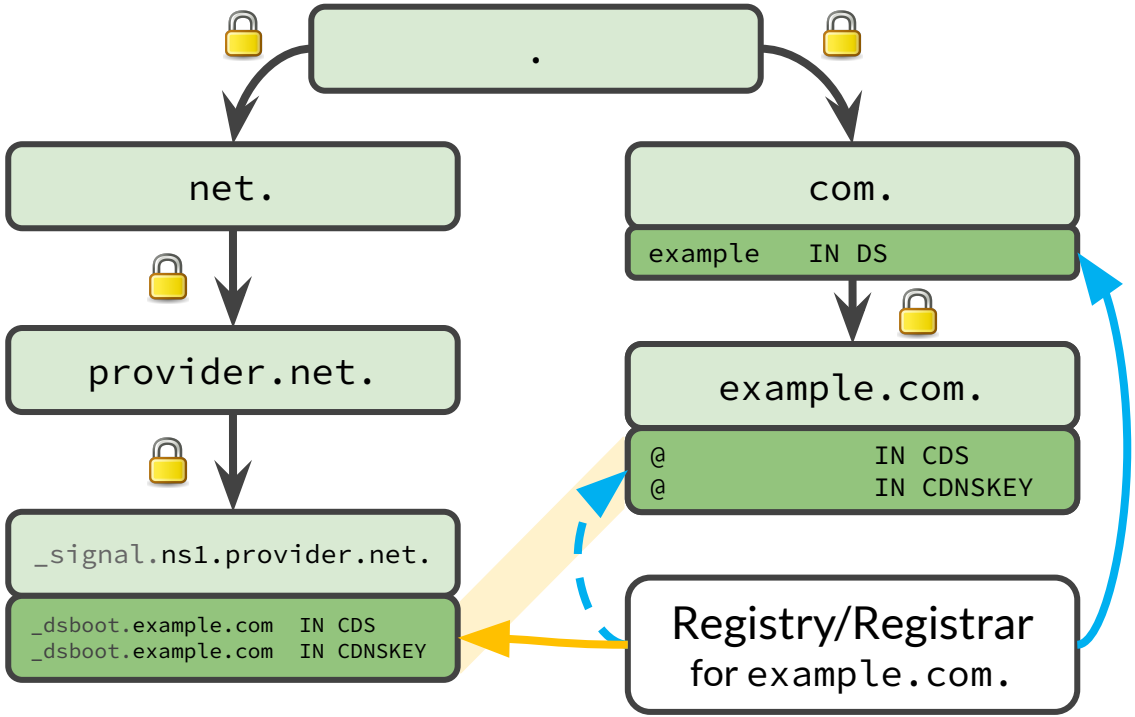
- No manual dealing with cryptographic parameters
- Authentication? 🤔

CDS/CDNSKEY Processing from Insecure Child



- Accept after delay (RFC 8078 Section 3.3)
- Trust on first use (“TOFU”)

CDS/CDNSKEY Processing with Authentication



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

- identically co-published
- authenticated, immediate
- no active on-wire attacker

Extends RFC 8078 to add authentication for initial DS

Protocol Status

- Passed IETF Last Call and about to finish IESG review → RFC expected soon

- Parent side (processing signaling records):
  - Implementations at .ch/.li
  - Some gTLDs working towards deployment

- Child side (publishing signaling records)
  - Operators: Cloudflare, Glauca HexDNS, deSEC
  - Authoritative servers: **Knot DNS 3.3.5** (released) and **PowerDNS 5** (projected)

deSEC

Knot DNS Implementation: `authsignal` module

```
zone:

  - domain: example.com
    dnssec-signing: on
  - domain: …
  - domain: …
```

Knot DNS Implementation: `authsignal` module

```
mod-onlinesign:
  - id: authsignal
    nsec-bitmap: [CDS, CDNSKEY]

zone:
  - domain: _signal.ns1.provider.net
    module: [mod-authsignal,
             mod-onlinesign/authsignal]
  - domain: example.com
    dnssec-signing: on
  - domain: …
  - domain: …
```

deSEC

PowerDNS Implementation: domain metadata flag

- Implementation available, currently slated for PowerDNS 5
  - Still deciding between two config approaches
  - (Meanwhile, LUA available → backup slide)

```
export nshost=ns1.example.net
pdnsutil create-zone _signal.$nshost $nshost

1)  pdnsutil secure-zone _signal.$nshost
    pdnsutil set-nsec3 _signal.$nshost "1 0 0 -" narrow
    pdnsutil rectify-zone _signal.$nshost
    pdnsutil set-meta _signal.$nshost SIGNALING-ZONE 1

2)  pdnsutil set-signaling-zone _signal.$nshost
```

Vendor-agnostic Implementation (no online synthesis)

- Some auths don't do online synthesis → requires static signaling zone
- `dsboot generate` is a generator for this purpose
  - Accepts NS records or CLI arguments to determine names of signaling zone
  - Accepts CDS/CDNSKEY records, constructs signaling records by their owner name
  - `-w`: write signaling zones to `_signal.$NS.zone` (or stdout); `-r`: read zonefiles for update
  - Available on PyPI in June (needs docs, tests)

```
dsboot_generate -r -w ns1.desec.io ns2.desec.org <<<EOF
thomassen.io 3600 IN  CDNSKEY  257 3 13 24w5jL63Q4b+buCmSiR6ZZ4UPcbZXA…
thomassen.io 3600 IN  CDS  50421 13 4 04e0cf24d5dfdcad3bcf2f1f400bbabb7ee3…
thomassen.io 3600 IN  CDS  50421 13 2 bb207d421bf9f4ec7ae976c1886b23da3e09…
deleted.test 3600 IN  NS   ns1.desec.io.
deleted.test 3600 IN  NS   ns2.desec.org.
EOF
```

Upcoming developments: DS Notifications

- CDS/CDNSKEY processing usually implemented via daily scan
  - Inefficient: lots of traffic, but things rarely change
  - Uncertain timing
- Better: have the operator of `example.com` send NOTIFY to the parent
  - But: where exactly?
- Look for DSYNC record at `example._signal.com` (or `_signal.com`)

```
example._signal.com. IN DSYNC CDS 1 51 target.registrar.
```

- Can point to endpoint run by registry or registrar
- draft-ietf-dnsop-generalized-notify (plan to implement at IETF hackathon)

Thank you!

… also to our sponsors:

Questions?

PowerDNS Implementation via LUA Interface

1. Enable LUA in config: `enable-lua-records=yes`
2. Download synthesis script:

```
curl https://raw.githubusercontent.com/desec-io/desec-ns/f3b561d43213d9e2b5c82dc7c2a8ed1fe7ea198d/ns/lua/signaling.lua \
  > /usr/share/lua/5.1/authsignal.lua
```

3. Hook up to signaling zone:

```
ZONE=_signal.ns1.provider.net  # assuming zone already exists
pdnsutil replace-rrset $ZONE * LUA 1 \
  CDS     ";require('authsignal') return signal('$ZONE', 'CDS')" \
  CDNSKEY ";require('authsignal') return signal('$ZONE', 'CDNSKEY')"
```