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

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

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

- authenticated, immediate
- no active on-wire attacker

CDS Authentication: Co-Publish under Trusted Hostname

Registry/Registrar for example.com.
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
    - whether the domain itself is secure (has validation path),
    - all NS targets in the delegation,
    - 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

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

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

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?**

**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?