Automated Delegation Sync from Child to Parent

Bootstrapping Delegation Mgmt in non-DNSSEC Environments

Johan Stenstam

March 19, 2024

Problem Statement

The two drafts draft-johani-dnsop-delegation-mgmt-via-ddns and draft-ietf-dnsop-generalized-notify and describe mechanisms for how to automate (and very much speed up) synchonization of delegation information from the child to the parent

- the `NOTIFY` mechanism triggers a lookup of data by the parental agent and therefore require the child to be DNSSEC signed.
- the `DNS Update` mechanism does not.

In the `NOTIFY` case there is **no bootstrapping issue to solve** (if the DNSSEC signature chain is broken, there will be no synchronization)

In the `DNS Update` case, **there is a bootstrapping issue**, because there is no signature chain to leverage from.

Problem Statement, cont'd

Therefore the problem statement becomes:

How should the SIG(0) public key be communicated from the child to the parent "agent" in a way that enables the parent agent to trust the key?

The primary use cases are smaller scale operations, where the security models today are, in most cases, worse than what this will be.

But, even so, it would be great if it is possible to achieve both of

- improved security model for the bootstrapping case
- complete automation of the child—parent synchronization of delegation data

Model #1: Publish the public key in the child zone

- Publish the `KEY` record as it is, without a DNSSEC signature:

```
child.parent. IN KEY 512 3 15 Lw7kWxW713KCcT8p2FjtqIgIcRz/XY8wQRX6FR9Tqew=
```

- **Pro:** Trivially simple.
- **Con:** Not secure.

Model #2: Publish and confirm the key

- **1.** Publish the `KEY` record as it is, without a DNSSEC signature:

```
child.parent. IN KEY 512 3 15 Lw7kWxW713KCcT8p2FjtqIgIcRz/XY8wQRX6FR9Tqew=
```

- **2.** Parent require the child to explicitly confirm the key (via a web interface or whatever).

- **Pro:** Simple.
- **Con:** The child operator will in many cases just click "sure, that's my key" without checking. . . still not secure.

Model #3: Send a self-signed update requesting "upload"

- **1.** Publish the `KEY` record as it is, without a DNSSEC signature:

```
child.parent. IN KEY 512 3 15 Lw7kWxW713KCcT8p2FjtqIgIcRz/XY8wQRX6FR9Tqew=
```

- **2.** Send a DNS Update, requesting upload of the SIG(0) public key and sign the update with the private key ("self-signed").

```
ADD: child.parent. IN KEY 512 3 15 Lw7kWxW713KCcT8p2FjtqIgIcRz/XY8wQRX6FR9Tqew=
```

- **3.** Verify that **1** and **2** match.

- **Pro:** Still simple.
- **Con:** Slightly less bad, as an attacker would have to both spoof the `KEY` record and also generate the DNS Update. . . but that's doable, so still quite bad.

RFC 8078 To The Rescue

RFC 8078 deals with a similar problem: how to establish initial trust in a previously unsigned child zone by examining the child's published `CDS` record.

In section 3 of RFC 8078 a number of possible alternatives are listed, including "check several times over extended time", "check from several vantage points", etc.

Similar logic can be applied in this case to improve the security of the initial trust establishment.

In Some Cases Explicit Confirmation **Will** Be Needed

The problem with explicit confirmation of a key is that many children will have no idea how to check the correctness of the key.

- . . . so there is a real risk that they will just click "yes" to make the question go away.

So let's ask for something simpler. Like the IP-address.

If the DNS Updates are always **sent over TCP** to the parental agent then it will be more difficult to forge the source address of the update.

- Whoever is authorized to confirm the authenticity of the SIG(0) public key upload request should reasonably be able to vouch for the correctness of the source IP address used.

Proposal

- **1.** Publish the `KEY` record as it is, without a DNSSEC signature:

```
child.parent. IN KEY 512 3 15 Lw7kWxW713KCcT8p2FjtqIgIcRz/XY8wQRX6FR9Tqew=
```

- **2.** Send a DNS Update **over TCP**, requesting upload of the SIG(0) public key and sign the update with the corresponding private key.

```
ADD: child.parent. IN KEY 512 3 15 Lw7kWxW713KCcT8p2FjtqIgIcRz/XY8wQRX6FR9Tqew=
```

- **3.** When the update is received the parental agent should look up the `KEY` via direct query to the child authoritative nameservers and verify that **1** and **2** match.

  Do this **N times**, spaced apart so that a possibly spoofed `KEY` record will expire between rounds.
- **4. If or when needed:** Require the child operator to verify that the update is sent from the correct source IP address.

- **Pro:** Not too difficult. **Con:** Without the explicit confirmation there is still a window of vulnerability. . . but it is getting quite small.

Implementation Status

Most of this is implemented, during the Hackathon and afterwards during this week.

- The internet-draft will be updated shortly.
- Both the "**generalized notify**" scheme and the "**DNS Update**" scheme are being implemented.
  - Both "child side" and "parent side".
- Planning to have a fully functioning implementation that completely automates synchronization of delegation information ready later this year
  - Possibly in time for IETF120.
- The implementation is open source.