DNS-SD SERVICE REGISTRATION

Ted Lemon <mellon@fugue.com>
Stuart Cheshire <cheshire@apple.com>
STATUS

➤ Document was expired

➤ Update was posted prior to this IETF (-01)

➤ Discussion ensued on mailing list (thanks, Toke!)

➤ Tim Wattenberg did a service implementation

➤ Second update, posted to IETF Monday (-02)

➤ A ton of discussion after that, being tracked on github

➤ Call for adoption is underway

➤ Document is actually in pretty good shape

➤ Has been thoroughly reviewed
WHAT IT DOES

➤ Provides a lightweight process services can use to register in the DNS

➤ Provides first-come, first-served protection for naming

➤ Provides garbage collection for
  ➤ Claimed names (14 days?)
  ➤ Service registrations (2 hours?)

➤ Constrained devices update to Anycast UDP or TCP

➤ Less-constrained devices discover dnssd-srp service and send updates to it using TCP
ISSUES

➤ This uses DNS update, but requires custom semantics

➤ This is required because we are allowing unauthenticated devices to register

➤ By tightly constraining what can be in a registration, we prevent arbitrary publication of names

➤ These semantics have to be implemented by the server that processes the update, so either you have a DNS server with some heavy custom semantics, or you need a shim between the authoritative server and the SRP service

➤ I don't think there's a way around this that allows ad-hoc registration, which is an obvious requirement
USE OF .SERVICES.ARPA

➤ Anycast Registrations update .services.arpa.

➤ This is not where the registration will actually go—it will go to dr._dns_sd.<domain> or x.y.z.q.in-addr.arpa or a.b.c.d.q.o.m.g.s.o.m.a.n.y.d.i.g.i.t.s.ip6.arpa.

➤ Semantics of a DNS Update include that it updates a single zone

➤ We can either violate that semantics or require that the update go to xxx.in-addr.SERVICES.arpa and xxx.ip6.SERVICES.arpa.

➤ Are we okay with this? Which should we do?
DOES NOT SUPPORT INTERNAL NATS

- A Registration for an IPv4 address will only be reachable if
  - the IPv4 address is global or
  - the user of the service is in the same RFC1918 routing domain

- I think this is okay

- A really badass registration server could set up an external SRV and a PCP port mapping, but that's another document.
Do we want to require that the update be for the address it came from?

- If so, then if a service wants to support dual-stack, it does two updates
- If a service has a ULA and a GUA, it has to pick, or do two updates

Should we give advice about this? e.g.

- If there is a ULA, use that by default
- If configured for public access, use GUA if present
- If only GUA present, use that?
- What if there's more than one ULA or GUA?

Alternative: let hosts update all addresses at once

- Is that actually better?
- What are the risks?
ONLY DNS-SD RECORDS SUPPORTED

➤ Very restrictive about what constitutes a Registration
➤ Service Name: only PTR, no delete
➤ Service Instance Name: only SRV and TXT
➤ Forward Mapping: only A or AAAA, plus required KEY
➤ Reverse Mapping: only PTR
➤ Service Name must point to Service Instance Name in update
➤ Service Instance Name SRV must point to Forward Mapping in update
➤ Reverse Mapping must point to Forward Mapping
➤ Benefit: we don't allow random updates
➤ Disadvantage: we don't allow random updates
➤ What about simple hostname updates? Allow or not?
The idea is that the stateful part of the service is not on the local network.

This means that for RFC1918 addresses, IP source address validation isn't going to work end-to-end.

To make this work, I think that you need a (mostly) stateless relay on the local network which validates the Registration and then uses TSIG or SIG(0) with its own key to do regular RFC2136-style updates to the cloud server.

Nothing technically hard about this, but do we need to specify it?
TOKE'S CLOUD SERVER, TAKE 2

➤ If we want public services,
  ➤ combine this with PCP
  ➤ cloud update points to PCP-assigned port on home router
  ➤ which is mapped to the internal IP address of the service
  ➤ now the service is publicly reachable
  ➤ still requires a relay
➤ Do we care about this use case?
➤ Why not just use IPv6? :)

10
The document explains how a service can register using plain DNS Update if SRP is not available.

It also talks about how to use a plain DNS Update server to test SRP in the absence of an SRP server.

Do we care about this?
Discovery Proxy assumes one subdomain per link
Registration protocol has no such requirement
Therefore, that's yet another subdomain
Right?
Thotz?
DELETION

➤ Current spec assumes that records are garbage collected and never deleted

➤ If a device changes its name, that could take a while to look pretty again

➤ Should we also allow deletes?
WHAT ABOUT SHARING NAMES ACROSS DEVICES

➢ Do we address this use case?
➢ Use a common key between devices?
➢ Some other thing?
NEXT STEPS

➤ Despite being in CFA, I think document is actually nearly ready to publish

➤ If you don't think that, or are skeptical, please review and send comments

➤ I would like to move quickly with this

➤ What do you think?