Multicast DNS Discovery Proxy

Ted Lemon <ted.lemon@nominum.com>
Stuart Cheshire <cheshire@apple.com>

Status Quo

Discovery Proxy does:

- Receive queries for per-link zones, e.g.:
  - _ipp._tcp.link-1.home.arpa IN PTR?
- If we have live answers in cache, send them to requestor
- Translate per-link zone name to .local in name(s) being queried, e.g.:
  - _ipp._tcp.local IN PTR?
- Construct new query that mentions what is already cached to avoid unnecessary repeats

Status Quo (continued)

- Send those queries on the link specified by the zone
- Listen for mDNS responses
- For each response received:
- Translate:
  - the name in the response from .local back to per-link name
  - any names ending in local in RRsets to the per-link name, e.g.:
  - _ipp._tcp.local IN PTR printer-1.local -> _ipp._tcp.link-1.home.arpa IN PTR printer-1.link-1.home.arpa
- Send response to querier
- Cache response in case of similar later queries

Observations

- Discovery proxy, done right, is fairly heavyweight
- On a network with many links, many caches
- Many separate translators
- Distributed state creates management complications

Proposal

- Separate link-resident relay
- Relay is stateless: no cache, no translation
- Discovery Proxy service can be centralized
- Discovery Proxy can still be distributed
- Relay is essentially a virtual interface for mDNS

Details

- Discovery proxy is essentially unchanged, except:
  - to do mDNS on a particular link it may:
    - speak directly to link, of connected to link
    - speak to link using relay when connected to link
    - speak to link using relay when not connected to link
- Discovery Relay does no translation
- Discovery Proxy and Relay communicate over TCP+TLS using pre-shared public/private keys.
- Discovery Proxy does all caching
- Discovery Proxy talks to resolvers; relay does not.

Management

- The draft goes into a lot of detail about how discovery proxies know about relay proxies and vice versa.
- This makes it look complicated, but for the most part it's not really complicated--it's just that the ops bindings are fully specified, and that's a fair amount of detail
- The specification is intended to work for:
  - manual configuration
  - management using netconf/yang
  - automatic management using HNCP/DNCP
  - Any other similar mechanism

Questions

- Put management bit in a separate document?
- Is TLS the right way to secure this?
- Do people think this is useful?
- Adopt?