Service Replication Protocol

Ted Lemon <elemon@apple.com>

But first, a quick roadmap

- mDNS is the original permissionless DNSSD
- DNSSD on DNS is the original infrastructure DNSSD
- Discovery Proxy lets us integrate mDNS into DNS infrastructure
- SRP lets us register services with a DNS authoritative server
- Advertising Proxy lets us advertise services in a DNS zone using mDNS
- SRP Replication lets us automatically manage a DNS zone in the absence of infrastructure support
- Permissionless Advertising of DNS Zones (PADZ) allows:
    - DNSSD clients to query SRP databases and Discovery Proxies without infrastructure support
    - Discovery Proxies and SRP databases to register with infrastructure where available

Why a new protocol?

- Could use authoritative DNS + IXFR, but problems:
    - The source of truth is really the SRP update, and we have no way to represent that in a DNS zone, so we'd need a bunch of new RRtypes which we'd then have to filter out of answers
    - We'd need a way to elect a primary when the primary goes away
    - Lots of corner cases during failovers that are messy to deal with

SRPL overview

- Arbitrary number of SRP replication peers (could limit it)
- No server is the authority
- Goal is to maintain a common set of SRP registrations (a dataset)
- Every server talks to every other server
- Because SRPL peers are peers, each has a precedence
- Servers with higher precedence numbers connect to servers with lower precedence numbers
- On connection, we do a database sync/reconciliation
- Then whenever an SRP update arrives, the server that received it replicates it to all the others

Why this works

- SRP clients are going to update a single server
- SRP clients are identified by a public key for which only they have the private key
- So we can tell whether an update is from the same client
- A more recent update should always be more correct than a less recent update from the same client
- So if we just effectively treat the client as the authoritative name server for its data, we can maintain an SRP dataset that is always roughly in sync, and always tending towards becoming more correct, not less

What is replicated

- The effect of replication is to replicate data in a zone, but what we actually replicate is the set of SRP updates required to get the dataset to the most recent state
- IOW, when an SRP server gets an update, it remembers the entire payload of the update and what time it was received, and propagates that to other SRPL peers
- Because we know when the update was received, we know whether it's more recent than what we have, and we can validate the signature even if it contains a timestamp

Protocol Flow: Discovery

- Choose a dataset
  - What dataset to choose is somewhat situationally dependent
- Discover SRPL peers replicating our dataset
  - If other SRPL peers are in the same situation, will want to replicate the same dataset
- If none, advertise precedence=1 for our dataset
- If >0, connect to the first and get a precedence number
- Then connect to the others

Protocol Flow: reconciliation

- Connecting peer runs through its list of SRP registrations
  - For each registration, proposes it to its peer, providing hostname and time-of-receipt
  - If its peer doesn't have it, or has a less recent version, it will request it; if requested, connecting peer sends it
- Accepting Peer runs through its list of SRP registrations
  - Same as above
  - We don't optimize out hosts touched in the previous update, because an SRP update might arrive during reconciliation
- This is a full database replication, including updates originally received from other SRPL peers, which might no longer be online

Protocol Flow: Normal Operation

- When a peer gets an SRP update, it replicates it to all its peers
- When any of its peers get updates, they send them to it
- No second-hand updates during normal operation

Current Status

- Two implementations:
  - Apple (tvOS 15)
  - OpenThread (current)
- No interop testing yet
- Apple implementation is behind the current spec ATM
- Would like the working group to adopt this work