Reliable SRP without an operator

Ted Lemon <mellon@fugue.com>
Jonathan Hui <jonhui@google.com>
Abtin Keshvarzian <abtink@google.com>

Problem Statement

Stub routers can't be assumed to always be present

Stub routers may need to provide SRP authoritative service

In some cases there will be multiple stub routers servicing the same stub network and the same infrastructure

We can use this to increase reliability by replicating the SRP authority data across multiple stub routers

How?

Givens:

- SRP Requestors are able to securely identify themselves
- SRP Requestors update a single SRP Registrar
- Therefore, the most recent update from the Requestor can always be assumed to represent the current state of the SRP registration
- Source of authority is thus the Requestor, not the Registrar
- Since there's provably one Requestor for any given SRP registration, we can rely on this to maintain the "authoritative" data in a shared database without fancy DBMS fu
- This lets us easily tolerate loss of any particular server

Documents

draft-lemon-srp-replication

- The actual SRP replication protocol

draft-tllq-tsr

- An mDNS change to prevent temporary conflicts

Status of Replication

SRP Replication database update protocol is mature

- Lots of discussion on mailing list
- Google and Apple both have implementations
- No interop testing yet

Recently changed the way SRP starts

- Goal: prevent multiple SRP datasets per network

Issues

The way datasets are named is poorly specified

- Possibly this is the wrong document for this anyway

We don't currently have a protocol version number

- This has been an issue because message formats have changed as the discussion has proceeded
- Should we have one?

Current implementations are using experimental/local-use DSO type codes

- Would be nice to get an early allocation

Document is not actually a WG document

- Can we adopt? Encourage more discussion?

What is TSR?

When an SRP update happens

- SRP server publishes new data
- SRP server replicates data to other replication peers

Problem:

- New data conflicts with old data
- Name conflict detected
- We either ignore it and wait for the last server to stop advertising the stale data, or
- detect a conflict (every time!) and tell the client to rename

This is because mDNS assumes incumbent registration is authoritative.

Solution

Get rid of “incumbent is authoritative” assumption for proxies

- Non-proxy registrations do not include TSR record
- Proxy registrations do include TSR record
- TSR record says “this is how long ago I got this data.”
- If a conflict is detected, and TSR is present both in the authoritative data and the mDNS data, we prefer the more recent data, rather than the older data.

This preserves old behavior, but fixes things for proxies

Status of TSR

TSR has been implemented at Apple

- We learned a lot from the implementation
- tsr-02 draft reflects lessons learned
- Fair amount of discussion among implementors on list
- Document is fairly mature: no known outstanding issues
- Would like to get more WG review to see if we missed anything
- Therefore would like WG to adopt the document

Thotz?