DNS-SD Privacy
draft-ietf-dnsssd-privacy-01.txt

Daniel Kaiser, Christian Huitema
IETF 98
March 28, 2017
Since previous drafts

• Lots of work on implementations
• Simplifications
• API issues
Simplified instance names from draft 00

• Version 00: Instance name for list of peers
  • Name = <nonce><proof₁>...<proofₙ>

• Version 01: One instance per pairing
  • Name 1 = <nonce><proof₁> (pairing 1)
  • Name N = <nonce><proofₙ> (pairing N)

• Rationale:
  • Complex name was really an “early optimization”
  • With domain name compression, >50 records in 1500 byte message
Standardized on nonce = 24 bit time

- Mitigates potential DDOS attack
  - On new nonce, need to compute proofs for all pairings
- Nonce = most significant 24 bits of 32 bit time
  - New computation at most every 256 seconds (4 minutes and 16 sec)
    - Frequent enough to mitigate replay attacks.

- Issue: on DNS based deployments, requires updates every 4 minutes
  - Do we need a special case?
Short proofs, BASE64 encoding

• Proof = SHA256(<nonce>|<pairing key>)
• Instance Name = BASE64(<nonce>|<proof>)
• What length?
  • Large enough to prevent name collisions => at least 32 bits
  • Short enough to generate short instance names
  • BASE64 uses 3 bytes for up to 24 bits
• Decision: 24 bit nonce, 48 bit proofs, 12 characters instance names
  • If using BASE32, would need 16 characters, 24 + 56.
Direct queries

• Client can predict the “hint” used by the peers
  • Assume better than 1 minute time synchronization

• Client can compose list of “potential instance names”
  • Instance Name 1 = <nonce><proof_1> (pairing 1)
  • Instance Name N = < nonce><proof_n> (pairing N)

• Client can send multiple queries for <instance_i>_pds._tcp.local

• DNS-SD allows us to pack several queries in one request
  • > 50 queries fit in single message

• Unicast responses are just fine, and lower overhead
“private” subdomain for private discovery?

• Alice looks for “_example._tcp.private.<domain>”
• Resolver first looks for “private discovery server” of every friend
  • Directed discovery:
    • For each friend in table, computes hash(seed, Friend’s PSK)
    • Sends request for “<hash> || <seed>._psds._tcp.<domain> SRV IN”
    • When using mDNS, each packet can contain several queries
  • Or, Global discovery, using DNS-SD:
    • Sends request for _psds._tcp.<domain> PTR IN”
    • Gets lots of PTR <hash> || <seed>._psds._tls.<domain>
    • Filters those that come from friends
    • Resolves SRV, etc.

• Resolver sets DNS over TLS connections to each private server
  • Forwards DNS requests, perform DNS-SD discovery, etc.
Next steps

• Implementation in GetDNS
  • In progress, Christian Huitema

• Some discussion...

• Last call?