DNSSD Privacy Scaling

Christian Huitema

IETF 101, March 2018

DNSSD Privacy Draft Feedback

- Very little actual feedback, except at IETF 100
- Several issues
  - Relevance, lack of integration with new IOT standards
  - Per device model, versus per application
  - Details of pairing process
  - Scaling
- In this presentation, focus on scaling
  - Tradeoff between scaling and privacy

Secrets for privacy, 3 options

- Client-server pairing (DNSSD privacy & pairing drafts)
- Secret shared by all clients
- Public key of the server
  - Basic design:
    - query with hash(key, nonce)
    - Reply with proof(nonce, private key)
  - Assume that the public key is only known by authorized clients
  - Key is unique ID of server
  - If known by adversaries, then adversaries can track the server

Scaling properties

N	Number of clients per server
M	Number of servers per client
P	Number of servers present in scope

	Pairing	Shared secret	Secret public key
Number of records published per server	O(N)	O(1)	O(1)
Number of responses per query	O(N*P)	O(P)	O(P)
(Query with secret dependent service ID)	O(1)	O(1)	O(1)
Optimized queries per client (DNSSD privacy)	O(M)	O(M)	O(M)
Caching possible (MDNS style)	No	Yes	Yes

Privacy Properties if client is compromised

	Pairing	Shared secret	Secret public key
Discover peered servers	Yes	Yes	Yes
Discover other clients of servers	Maybe	Yes	Maybe
Impersonate peered servers	No	Announce	Announce
Cost of remediation	O(M)	O(M*N)	O(M*N)

- Without compromise, everything works
- Server compromise has same effects for all solutions

Scaling and Privacy Properties

	Pairing	Shared secret	Secret public key
Number of responses per query	O(N*P)	O(P)	O(P)
(Query with secret dependent service ID)	O(N) pub, O(1) replies	O(1)	O(1)
Resist compromise	Yes	No	Maybe
Client compromise remediation	O(M)	O(M*N)	O(M*N)

Christian's preferences

- Without constraints
  - Move to "secret public key" class of solution
  - Use "hash of public key and nonce" as service identifier
  - Use "proof of nonce with private key" in response
  - Tie with use of PSK and public key in TLS (TBD)
  - But "proof with public key" does not fit in 64 characters
- If we must keep the DNS-SD protocols and formats
  - Keep "pairing secrets" for short message size, compromise remediation
  - Consider per application simplification
  - Use secret dependent service type (randomized service type)