Next Steps for DNSSD Private Discovery

Christian Huitema

IETF 103, Bangkok, November 2018
## Summary of “privacy scaling” draft

<table>
<thead>
<tr>
<th>Solution</th>
<th>Scaling</th>
<th>Resistance</th>
<th>Remediation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pairing secret</td>
<td>Poor</td>
<td>Bad</td>
<td>Good</td>
</tr>
<tr>
<td>Group public key</td>
<td>Medium</td>
<td>Bad</td>
<td>Maybe</td>
</tr>
<tr>
<td>Shared symmetric secret</td>
<td>Good</td>
<td>Really bad</td>
<td>Poor</td>
</tr>
<tr>
<td>Shared public secret</td>
<td>Good</td>
<td>Bad</td>
<td>Maybe</td>
</tr>
</tbody>
</table>

Ref: draft-ietf-dnssd-privacyscaling-00
Shared Public Secret Solution

- Server has a public key & private key pair
- Authorized clients learn the public key
  - They keep it secret!
- Discovery request demonstrates knowledge of public key
  - Nonce + Hash (nonce | public key)
- Discovery response demonstrates ownership of private key
  - E.g. TLS connection handshake
- Scaling: 1 request & 1 response per server
- Resistance: if public key disclosed, server can be discovered
- Remediation: revocation of old key, distribution of new one
Predictable Nonce

• Nonce = Quantized Time
  • E.g. most significant 24 bits of Unix 32 bit time
• Server/Client needs to compute Nonce + Hash once per epoch
• Enables pretty good scaling
  • Client can compute nonces of all interesting servers
  • Servers publish record per nonce (DNSSD) or filter per nonce (MDNS)
  • Server records can be cached (DNSSD, MDNS)
Example Discovery, MDNS

Client computes Nonce, Hash (Sends MDNS request: base64(Nonce|hash).local ➔

➔ Server Recognizes base64(Nonce|hash)
Prepares signature (Nonce, hash, randomized name)
← Sends MDNS reply: Nonce, randomized name, signature

Receives reply ←
Verifies signature
Starts TLS connection

Other clients may cache the response
Question: Keep the DNS formatting?

• DNS Format:
  • Format (nonce|hash) as “service type”
  • Format signature as “service name” (maybe) or TXT record

• DNS format pros & cons:
  • Reuse MDNS networking code
  • Reuse DNSSD servers, proxies, etc.
  • Awckward compromises, e.g. size of hash, size of signature, service type

• Binary alternative:
  • More natural encoding
  • But yet one more multicast based discovery...
Reconcile with Bob Bradley’s proposal

• Bob’s proposal = mirror image of “server public key”
  • Query: signed with client’s public key
  • Server processes query, check whether signed by authorized client
  • Response if client is authorized

• Advantage of Bob’s proposal:
  • One query from the client, responses from every server present
    • Client does not need to send one query per server.

• Drawbacks of Bob’s proposal:
  • Hard to reconcile with DNSSD “server mode”
  • Requires “trial decryption”, potential DOS on servers
    • Could be mitigated by adding “predictable nonce” to the query
Next Steps?