XMPP DNA

Richard Barnes

Stephen Farrell

Problem

- Example.com outsources XMPP services to example.net
  - SRV _xmpp._tcp.example.com → example.net
- Want example.net to be able to authenticate as himself, not example.com
- Need to secure delegation: How do I know that example.com really delegated to example.net?

Solution approaches

- Attribute cert:
  - CMS encoding, CMS signature
- ~~XML Assertion:~~
  - XML encoding, HTTPS signature
- DNS Assertion:
  - SRV encoding, DNSSEC signatuere

Observation: DNSSEC fixes things

- If the SRV is signed in a way that the client can verify it, then there's no problem
- Client needs an appropriate trust anchor
  - Root, DLV, ITAR, etc.
- Problem arises when the client doesn't have a TA that can be used to validate the signature

Bridging the gap

- If the problem is a missing TA, supply it at the application layer
- Need to bind a name to a key
  - … so supply a certificate under a well-known CA

```
<challenge>
  <proof type="urn:ietf:params:dna:proof:dnssec-ta">
</challenge>
<proof>http://example.net/example.com.cert</proof>
```

Overall Process

1. Want to send a message to example.com
2. Look up SRV for target domain, get example.net
3. If DNSSEC-secured, match against example.net
4. If not, challenge server to provide a TA / cert
5. Validate SRV under that TA
6. If you don’t get a secure delegation, FAIL
7. If the genuine delegate isn’t example.net, FAIL
8. Otherwise, SUCCESS

Pros & Cons

- Pro: Forward-compatible with DNSSEC
- Pro: Doesn't require attribute certs
- Pro: Only requires outsourcing provider to have a cert for the outsourced domain (not a private key)
- Con: Requires application control of DNSSEC TAs
  - But: `ub_ctx_add_ta(struct ub_ctx* ctx, char* ta);`
- Con: Requires binding between certificate and DNSSEC key pairs

Open issues

- Feasibility of managing DNSSEC TAs
- Feedback from DNSSEC community
- New approach for draft-ietf-xmpp-dna?