DANE – a killer app for DNSSEC?

CZ.NIC z.s.p.o.
Ondřej Surý
ondrej.sury@nic.cz
27. 6. 2012

cz nic
cz domain registry

Certificates in DNS

- Old and recurring idea...
- CERT RR (RFC4398 by Simon Josefsson)
  - Generic way to store certificates

  Hostname/Email **CERT** Type KeyTag Alg Cert_or_CRL
- draft-schlyter-appkey (by Jakob Schlyter

  Hostname **APPKEY** PubKeyAlg PubKey
- TLSFP RR request (by Ondřej Surý
  - many comments → withdrawn to the date

  _Service._Proto.Name **TLSFP** Port Mandatory PubKeyAlg HashAlg FingerPrint

What we do in DANE WG?

- Put a {key|cert|hash|...} to DNS
- Sign it with DNSSEC
- Use that as a “trust path” for certificates

Why we do it?

- DNS lookup + TLS negotiations
  - Can take a long time (OCSP, CRL, …)
- DV certs – heap of unknown CA trusted
  - If you trust one, you trust them all
    - Government CAs...? (
    - Wildcard ('*') certificates
    - CA breaches (remember DigiNotar?)
- Solution: Use DNS to publish the “correct” key for the host
  - Can (even) save time (do DNS lookup in parallel)

Where we were?

- Idea sort of floating around for a long time
- DNS root got signed!
- Bar BOF @ IETF 78 in Mastricht (2010)
  - Mailing list created (keyassure)
  - Lots of discussion
  - 5 new Internet Drafts so far
- BOF @ IETF 79 in Beijing (KIDNS BoF)
  - Working Group created few weeks after that
- WG renamed to DANE (to not clash with kitten)

Where we are?

- Working Group Documents
  - draft-ietf-dane-use-cases
    - Describe use cases for DANE
    - RFC 6394
  - draft-ietf-dane-protocol
    - Two IETF Last Calls
      - The first one generated lot of comments
    - Got approved by IESG in June 2012
    - **In RFC Editor queue! Hooray!**

Use cases in a nutshell

- CA constraints
  - “I use only this CA for my certificates”
- Certificate constraints
  - I use only this CA-issued certificate”
- Domain-Issued Certificates
  - “I have generated this certificate and I use it”
- Delegated Services
  - “My hosting provider has to use this certificate”
- Web Services
  - “Machine-to-machine communication”

What does TLSA look like?

- Query: **_portnum._prottype.hostname**
  - 1 query → *n* responses (rollovers, load-balancing)
- Response:
  - cert_usage selector matching_type binary_data
- Example:

```
_443._tcp.example.com 3 1 1
8755CDAA8FE24EF16CC0F2C9180631 85E43
3FAAF1415664911D9E30A924138C4
```

What does TLSA look like?

- Certificate usage:
  - 0: CA constraint
  - 1: Service certificate constraint
  - 2: Trust anchor assertion
  - 3: Domain-issued certificate
- Selector
  - 0: Full certificate
  - 1: SubjectPublicKeyInfo
- Hash-type:
  - 0: Full certificate
  - 1: SHA-256 hash
  - 2: SHA-512 hash

Certificate usages

CA constraint	Service certificate constraint
• CA certificate • MUST pass PKIX validation • „use only this CA“	• End-entity certificate • MUST pass PKIX validation • „use only this cert from CA“
Trust anchor assertion	Domain-issued certificate
• Self-issued CA certificate • Insert new trust anchor • „use my own CA“	• Self-issue EE certificate • Must match service cert • „use my own certificate“

TLSA record

- Already assigned by IANA
  - RR type 52
- Support for TLSA record in:
  - Bind 9.6-ESV-R7, 9.7.6, 9.8.3 & 9.9.1
  - Knot DNS 1.0.4
  - PowerDNS 3.1
- Generators
  - swede (https://github.com/pieterlexis/swede)

What is needed now?

- Patch the apps to support DANE
  - DNSSEC validation (or use trusted last mile)
  - Implement DANE matching
  - Browsers, MUAs, MTAs, XMPP, ...
- Fix the broken last mile
  - Dumb resolvers, captive portals, etc.
  - DNSSEC-Trigger can help here?
    - http://www.nlnetlabs.nl/projects/dnssec-trigger/

What to do next (protocol wise)?

- DANE and other protocols
  - S/MIME (draft-hoffman-dane-smime)
  - SMTP (draft-fanf-dane-smtp)
  - XMPP (draft-miller-xmpp-dnssec-prooftype)

Questions?