How to use DNSSEC to keep PKI on a leash

Jakob Schlyter
ICANN’46 DNSSEC Workshop
¿ Jakob Schlyter ?

- DNSSEC since 1999
  - .SE and other ccTLDs
  - Root DNSSEC Design Team
- IETF contributor – SSHFP, DANE
Why does PKI have to be kept on a leash?
Problem #1

• Any universally trusted (e.g., WebTrust™) PKIX Certification Authority can issue a certificate for any host on the Internet.
  ▸ X.509 name constraints are not in wide use.
  ▸ CA malpractice hard to mitigate.
  ▸ Attacks happening today. Perhaps even here.
Problem #2

- The current CA model effectively puts a tax on enabling secure communications.
  - Domain validation certificates costs approximately $12 per host.
  - Extended validation costs considerably more.
Goal

• Enable path restrictions
  ▸ Limit the amount of damage that a single Certification Authority can do.

• DNSSEC for identity validation
  ▸ Certification validation without legacy PKI
If DNS is used for identity proofing ...
… and DNSSEC provides data origin authentication,
why involve a 3rd party?
Is this a new idea?
“Alternatively, if certificates are retrieved from a secure DNS zone with DNS security checking enabled and are verified by DNS security, the key within the retrieved certificate MAY be trusted without verifying the certificate chain if this conforms with the user's security policy.”
draft-schlyter-pkix-dns-02

2002

PKIX WG – “We are not amused”
DANE

DNS-based Authentication of Named Entities
RFC 6698

Provides bindings of keys to domains that are asserted by DNS
PKIX Hierarchy

Name: diginotar.nl.
PubKey: deadc0fee...

Name: www.example.com.
PubKey: 1337f00f...

DNSSEC Hierarchy

Name: (root)
PubKey: 088eb101...

Name: com.
PubKey: c0c0bebe...

Name: example.com.
PubKey: 4242eeee...
CA Lock

- TLSA enumerates acceptable CA certificates
  - Only accept certificates under a specific CA
  - Optionally used together with classic PKIX

✓ Protects against CA malpractice, e.g., fraudulently issued certificates.
Certificate Lock

- TLSA enumerates acceptable EE certificates
  - Only accept specific certificates
  - Optionally used together with classic PKIX

✓ Addresses the problem with fraudulently issued certificates.
Self-signed Certificates

- Possible with or without a private CA

✓ Enables TLS without depending on existing PKI infrastructure.
<table>
<thead>
<tr>
<th></th>
<th>PKI</th>
<th>DANE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication</td>
<td>DNS</td>
<td>DNS when used</td>
</tr>
<tr>
<td></td>
<td>for identity proofing</td>
<td></td>
</tr>
<tr>
<td>Revocation</td>
<td>OSCP / CRL</td>
<td>DNS</td>
</tr>
<tr>
<td>Validation</td>
<td>PKIX</td>
<td>PKIX</td>
</tr>
</tbody>
</table>
Got rough consensus.
Got running code?
Implementations

- DANE Utilities
  - https://github.com/pieterlexis/swede

- Postfix with DANE
  - https://github.com/vdukhovni/postfix

- Generic OpenSSL with DANE
  - Work in progress
Challenges & Future Work
Last Mile Security

- Is DNSSEC validation in the endpoint realistic?
- Can we get DNSSEC data to someone behind filtering infrastructure?
  - Is tunneled DNS a viable solution?
DANE for S/MIME

draft-ietf-dane-smime-01

- Binding an email address to a domain name
- Use TLSA-like mechanisms to validate certificate
jakob@kirei.se