draft-lamps-bonnell-keyusage-crl-validation

IETF 121

Corey Bonnell

Tadahiko Ito

Tomofumi Okubo

What are Indirect CRLs?

- Certification Authorities (CAs) can delegate the signing of CRLs to other entities. (Think of OCSP delegated responders, but for CRLs.)
- These entities are issued an end-entity certificate with the cRLSign bit asserted in the keyUsage extension.
- Indirect CRLs cover certificates not signed by the CRL issuer themselves.
- Support for processing indirect CRLs is not mandatory to implement by RFC 5280, but several libraries support.

Inclusion of the keyUsage extension

RFC 5280, section 4.1.2.3 says:

```
Conforming CAs MUST include this extension in certificates that
contain public keys that are used to validate digital signatures on
other public key certificates or CRLs.
```

- No requirement for (non-CRL issuer) end-entity certificates to contain the keyUsage extension.

Processing of the keyUsage extension

RFC 5280, section 6.3.3 says:

(f) Obtain and validate the certification path for the issuer of the complete CRL. The trust anchor for the certification path MUST be the same as the trust anchor used to validate the target certificate. **If a key usage extension is present in the CRL issuer's certificate,** verify that the cRLSign bit is set.

- No explicit requirement for validating the inclusion of the keyUsage extension itself.

The Scenario

- A CA issues a CRL issuer certificate to subject X with key A, explicitly stating cRLSign usage.
- The CA issues certificates with crlDistributionPoints referencing subject X as the CRL issuer.
- The CA issues another certificate to subject X with key B, but without any keyUsage extension. This can be an end-entity certificate of any type, such as S/MIME, document signing, etc.
- Subject X signs a CRL using key B (not certified for CRL signing) and publishes it at the specified distribution point.
- Relying parties validate this CRL successfully, as the validation algorithm does not explicitly check for the presence of the keyUsage extension.

- We were able to reproduce this issue with several widely available implementations.

Proposed fix

Change (f) from:

Obtain and validate the certification path for the issuer of the complete CRL. The trust anchor for the certification path MUST be the same as the trust anchor used to validate the target certificate. ~~If a key usage extension is present in the CRL issuer's certificate, verify that the cRLSign bit is set.~~

To:

Obtain and validate… target certificate. Verify that the keyUsage extension is present in the CRL issuer's certificate and verify that the cRLSign bit is set.

Next steps

- Maintainers of several libraries with indirect CRL support have been notified of this issue, and we have been told fixes are planned or have already been released.
- We would like to formalize this modification to the validation algorithm.
- Call for adoption?