IETF PARIS 2012 PKIX meeting about : “not issued” and RFC 2560

Denis Pinkas

Denis.Pinkas@bull.net

March 27, 2012

About using the certificate database

- RFC 2560 states:
  - If nextUpdate is not set, the responder is indicating that newer revocation information is available all the time.

- Does it mean that when nextUpdate is not set, the OCSP responder is using directly the database of issued certificates ?

- If this is the case, it should be said more explicitly in rfc2560bis.

- Now let us see which additional information could be added to take advantage of the database or a part of it (partial export) in two cases:
  - whether the serial number is known or unknown in the database.

The serial number of the forged certificate is unknown in the database

- How to react if certificate has been fraudulently issued, but is not present in the database of issued certificates ?

- At a coarse level, this situation might arise if:
  - the HSM (Hardware Security Module) has been used directly without being accessed by the genuine CA application, or
  - the back-up keys of the HSM have been used to reload a spare HSM.

- For these two cases, the CA key MUST be revoked.
- The situation may also arise if :
  - the CA software has been partially tampered, but the writing in the database of issued certificates couldn’t be made:
    - Note that the tampering may allow to fake one certification profile, a few of them or all of them. This distinction is important as explained later on.

The serial number of the forged certificate is known in the database

- However, if the certificate is a forged one, a way to counter that threat is to send back in the OCSP response a hash of the certificate that is in the database (no need to move to OCSP v2).

- It would only be efficient in the following situation:
  - a certificate has been fraudulently issued, but his serial number is already present in the database of issued certificates (the writing in the database of issued certificates was still not made).

Strawman proposal

- In order to achieve backwards compatibility, a primary status "not-issued" should not be added to "good", "revoked" or "unknown".
- To handle the case where the serial number of the forged certificate is unknown in the database, a secondary status meaning "not-issued serial number" should be provided in the response as an extension.
- To handle the case where the serial number of the forged certificate is known in the database, a hash of the certificate should be included in the response as an extension.

- This leads to the following proposal :
  - When nextUpdate is not set, then an OCSP responder MAY include a non-critical extension which will contain either:
    - a hash of the certificate, if the serial number is present, or
    - a secondary status meaning "not-issued serial number",

- Question: which hash function should be used to compute the hash ?
- Response: the same hash function that was used to sign the certificate.

The three cases

- The serial number of the certificate is :
  - unknown in the database:
    - the primary status should be “revoked”,
    - the secondary status should be “not-issued serial number”.
  - known in the database:
  - if the certificate is revoked:
    - the primary status should be “revoked”,
    - the secondary status should be the hash of the certificate.
  - if the certificate is not revoked:
    - the primary status should be “good”,
    - the secondary status should be the hash of the certificate.

Additional considerations

- As already said on the PKIX list, supporting the “not-issued” case might provide a false sense of security, thus the limitations must clearly be advertised.

- Two cases need to be considered :

  1. the CA has designated an OCSP responder, or
  2. the CA signs itself the OCSP responses.

- In the fist case, the proposal is efficient only if both the writing in the database of issued certificates couldn’t be made and the software allowing to create certificates for an OCSP Responder couldn’t been tampered, otherwise then the attacker could create a fake OCSP service.

- In the second case, the proposal is efficient only if both the writing in the database of issued certificates couldn’t be made and the OCSP responder software couldn’t been tampered.

Efficiency

- The proposed extension will be efficient as long as:

  1) the CA key has been used to create end-entities certificates, but the CA database has not been corrupted.

  2) the CA key has been used to create end-entities certificates, and the CA database has been corrupted, but it is still possible to discriminate between genuine and fake certificates.

  - Additional action: revoke the fake certificates.

- But the key point is : how can you really know that :
  - that the CA database has not been corrupted, or
  - that you can discriminate between genuine and fake certificates ?

- If you don’t know or if there is any doubt, or if other conditions apply (e.g. OCSP certificates may have been created), then the extension is inefficient and the only solution is to revoke the CA key.