CA:Recommendations for Roots

Home » CA:Recommendations_for_Roots

Content

The following fields are not really Mozilla's thoughts, more a thought experiment on which to consider for the future, or that which iang figured out. See also CA:Overview

1.A hierarchical structure of a single root with subordinate or intermediate roots is preferred.

◦That's the case for IGC/A domain : there is a single root managed par ANSSI, wich subordinates roots for each french ministry.

2.The (single top-level) root's public certificate is supplied for Mozilla's root list; the subroots are not.

◦OK.

1.The root is expected to be offline, the subroots online.

◦The IGC/A root CA is offline, the immediate subordinate roots are offline too, and their subCA are online.

1.The audit process should cover the entire set of hierarchy.

◦That's the case. The audit process controls that no sub CA can be created without the agreement of the top level manager of the subordinate root CA ; ANSSI must also be informed.

Contents of the Root

Field	Constraint	Comments

O, OU, CN	no special constraints	Firefox displays O, Thunderbird displays CN
		O = ANSSI
		CN = IGC/A AC racine Etat francais

Validity	minimum 8 years, maximum 30 years	Minimum 10, maximum 18.
Format	x.509 version 3 PKCS1	OK
Public Key Algorithms	RSA	minimum 2048, maximum 4096 ... Consult [1] or [2]

		RSA4096 for all CA (roots and sub)
	ECC	256? 512? NO ECC for now
Message Digest Algorithms	SHA1	ok with NSS SHA1 is bannished
	SHA256	entire SHA2 family accepted by NIST SHA256
Certificate Policies (extension)	no limits	where any legal notices should go
subjectAltName	email address	contact for support purposes, if desired Generally not used.
pathLenConstraint basic constraints extension	pathLenConstraint=**n**	means, n of intermediates. where not present, no limit imposed. *recommended as unlimited, leave this one out.* Pathlength=0 : no limit in the root, because the CA hierarchy is controled by the ministries, and audited.

Following are obligatory, see [3] for more details.

Attribute	Name	Criticality	Constraint	Comments
SKID	Subject Key ID extension	**Critical.** **NOT Critical (cf RFC5280) but mandatory**	sKID=*SHA1(public key)* *OK*	unique id of this root, see [4] for format.
cA	basic constraints extension	**Critical.** **OK**	cA=true OK	means, Is a Certificate Authority
KU	key usage extension	**Critical.** **OK**	keyCertSign and cRLSign only **Sometimes : digital signature also**	obligatory for roots. bits 5, 6 to be set.

Following should not be included:

- CRL or OCSP indicators (for EE certs)
  - OK : no CRLDP indicated in the selfsigned root CA
- old "Netscape" fields (mostly deprecated)
  - OK : no such fiels
- Extended Key Usages (EKUs) which are for certs, not roots
  - OK : No EKU.
- logotype is ignored by all browsers at the moment??

- ? can't understand the question

SubRoots

Subroots or intermediate certificates are signed by the top level root, but include CA obligatory flags as above. They are intended to do the detailed day-to-day online signing of end-entity (EE) certs.

Q: how to differentiate one subroot for one purpose from that of another?

It is mandatory (in compliance to the « Referentiel Général de Sécurité » – the national IT security reference book) to differenciate each purpose (authentification certificates for people, for servers, electronic signature for people, e-sign for servers, encipherment) ; each CA must have a dedicated certificate policy OID for each type of certificate isued.

Q: how does the above mesh with the [policy]:

> *13. In addition to the requirements outlined above, we also recommend that CAs consider using separate root CA certificates with separate public keys (or separate intermediate CA certificates with separate public keys under a single root) when issuing certificates according to different Certificate Policies, so that we or others may selectively enable or disable acceptance of certificates issued according to a particular policy, or may otherwise treat such certificates differently (e.g., in our products' user interfaces). We reserve the right to make this a requirement in the future, and to not include a particular CA certificate in our software products, to discontinue including a particular CA certificate, or to modify the "trust bits" for a particular CA certificate, based on the CA's practices in this regard.*

It seems that (a) this suggests using separate single level roots, and (b) suggests that Mozo will control the intermediates. Now, it may be that this is 3 year old writings, and the experience has led to a different path .. does the policy need updating?

- Mozilla can control only selected intermediate certificates of a specific root and not include the root in case this is required. This is relevant for CAs which sub/cross sign third party CAs but only part of the (third party) CAs comply the Mozilla CA policy. Eddyn 12:33, 3 October 2008 (UTC)

- The policy might consider a CA path length of 0 for sub ordinate CAs which are operated by third parties to the root. This is a requirement for the issuing EV certificate. Eddyn 12:36, 3 October 2008 (UTC)

AKI extension

The AKI extension should include only the KeyID of the root (or higher subroot) key. It should not include the issuer name or serial number.

OK=> IGC/A and its subCA are totally compliant ; this is verified each time IGC/A issues au new certificate.

- [The Mozilla CA Certificate Policy] warns against this in Point 4: *incorrect extensions (e.g., SSL certificates that exclude SSL usage, or authority key IDs that include both the key ID and the issuer's issuer name and serial number); or*

The reason for this is explained in [this thread.]

Why should the authority key identifier (AKI) not include both the key ID and the issuer's issuer name and serial number?

- When an (end-entity) certificate's AKI extension contains only the KeyID field, it is possible to renew the (Intermediate) CA Certificate and use it to validate the (end-entity) certificate's signature. But if the (end-entity) certificate's AKI extension contains the Issuer Name and Serial Number fields, it is *not* possible to use a renewed (Intermediate) CA Certificate to validate the (end-entity) certificate's signature, because the renewed (Intermediate) CA Certificate has the "wrong" serial number.
- Having both the key ID and the issuer's name and serial number in the AKI is allowed, but it is almost always a huge mistake to do so. CAs that make this mistake typically have to abandon and completely replace their entire PKI (entire tree of issued certificates) when a CA cert expires and its serial number appears in the AKI of other subordinate certs.
- Almost without exception, most CAs that make this mistake do so because they use OpenSSL, and virtually every OpenSSL cookbook web page on the internet shows all 3 AKI fields being used. This is also in the default certificate generation configuration file for OpenSSL.

Revocation of the Root

To assist CAs in disaster recovery planning, the following is anticipated as Mozilla's root revocation process.

- For a root listed in Mozilla's root list, file a bug to request the root be marked "untrusted".
    - Include evidence.
    - CA should establish a separate channel to confirm.
- Mozilla will create an Advisory and distribute a security update. This is guesstimated to take around 5 business days: 2 for the NSS team, then 2 for the application teams (FF/TB/SM).
- Once distributed, any certs chained off the root will also be untrusted.

Explanatory Notes

- This method has been written with a view to assist disaster recovery plans and audit needs.
    - There is currently no better method.
    - CAs should factor in their own response time.
    - Also, note that Firefox has a staggered update time, to avoid server congestion.
- Note that this has never been done.
    - The timeframe anticipated above is an estimate.
    - The process might change in any real event.
- CAs will need to contact other vendors individually. E.g., anything outside the Mozilla family:
    - software that uses NSS but isn't a product of Mozilla,

    - other libraries.
- Within the context of PKIX:
    - NSS follows PKIX.
    - There is no PKIX method for revoking a root, this is considered to be a business issue.
    - Root revocation is generally expected to be handled in a vendor-specific way. That is, there is no commonality.
    - CRL/OCSP will only revoke downwards.
    - Paul H reports that there may be work on this at TAMP. The Requirements might be more digestable.
    - It is possible to cross-sign roots in PKIX with a Mozilla-specific root, but Mozilla is not in the CA business.