Andreas Kuehne
trustable Ltd.
kuehne@trustable.de

Proposal 'inconsistent CA' (inCA)

Motivation :

Testing of PKI software is a challenging task. To setup a test scenario targeting for a complete code coverage each and every error path must be taken. If a certificate authority is involved in this setup, the complexity increases. Common CA usually refuse to produce arbitrary inconsistent data just for testing purposes. Even if a company decides to run a private CA for test reasons the available software doesn't support every required type of irregular data.
Due to experience in the DSS/X TC, especially with the 'verification report' profile the 'valid signature' deserves only few attention compared with all the failure dimensions (cert validity, hierarchy and restrictions, CRLs, OCSP responses and their certificate tree, attribute certificates, trust lists, ..). So the usual 20 % core functionality versus 80 % error handling code must be shifted towards error handling.

Problems :

- Special requirement set is in direct contrast to common solutions.
- Incomplete test coverage degrades product quality.
- Complex test setup delays implementations .
- Much work for one project / company / TC.

Proposal :

- Gathering requirements from all involved TC to build a list of requirements. The DSS/X TC can supply a base set.
- Modifying existing open source software to support the requirements.
- Running a publicly available instance of the 'inconsistent CA'.
- Publishing the modified source code and documentation to ensure maximum persistence of the service.

Advantages of an 'inCA' effort :

- 'Once and for all' solution.
- Low or no administration required even in case of many users.
- A commonly reviewed and proven environment eases implementation efforts.
- Very special target zone, commercial CAs don't loose a market.
- Official support by Oasis / IDTrust may payoff as this tool helps to increase awareness and visibility the standard body.

So I would recommend that such an 'inCA' effort should be supported by Oasis / IDTrust. Defining a common 'inCA' requirement set is within the usual area of specifications done by Oasis. The deployment of an 'inCA' instance would help implementors a lot and could be seen as a specification support service.

Requirements driven by Verification purposes

The different aspects of a certificate verification can be separated into these major areas :

format compliance

Walk through the spec to identify MUSTS / MUST NOTs and construct a non-compliant certificate.

DER encoding problems

Negative exponent

chain integrity

Validity models and their violations

Issuing certificate identified by 'issuer' that doesn't matches the issued certificate.

Chains that never end (recursive issuer relationship).

Issuing certificates that are not intended for this usage.

Certificate chain exceeds given length limit

respect to constraints

End user certificates that are not intended for signing usage.

revocation information

Pointing to unavailable distribution points.

Distribution points returning invalid formats.

Distribution points don't sign the results properly.

Certificates inappropriate for the task.

Certificate chains invalid (see above).

attribute integrity

Invalid format of attributes.

Relationship of attribute certificate and certificate broken.

Timestamps broken (needs special considerations)

Certificate chains in attributes invalid (see above).

These aspects may again be applied to some of the sub-elements of some of some items. For example there is separate certificate chain to be verified when an OCSP response is checked. If we are going to generate a specific test certificate for each and every aspect there will be a big number of certificate to be produced. It may subject to special consideration whether permutations of 'defects' are required.

Looking at each of the topics above there are well known situations that must be represented by a test case, like known attack scenarios. On the other hand it's easy to get lost while constructing academic / exotic error scenarios. Obviously there are many more details to fiddle out to have at least a 'good enough' test set. It would be very useful to invite experts in these areas to share their knowledge and best practice in defining relevant error scenarios and use cases. See item one of the

proposal above.

Some of the requirements can be solved by configuring available software and its environment (like making OCSP responder unavailable), other need severe changes of CA software (like signing a certificate with another as the declared issuing certificate or applying a certificate for a usage that it's not intended for). CA software provider may refuse to introduce such dangerous code into their code base. A separate stand alone version may have the disadvantage of loosing the contact to the main branch development efforts.

Non-invasive deployment scenario

A major point of this proposal is the long term availability of this test environment. On the other hand it's not guaranteed that a given service provider will continue to offer its service for the next ten years. The obvious solution to do the hosting on an OASIS site brings the problems of administrating a complex third party product to an organization that's not in the business of running a CA.

So a mixed approach seems to offer the best benefit : OASIS holds the necessary URL like 'inCA.oasis-open.org' and forwards incoming requests to the inCA instance running at an arbitrary service provider. The forwarding of requests could be done with well known software like an apache http server or an openldap server. So there would be no need to run any new software on OASIS servers that the administrators are not already used to. This setup minimizes the security implications for the OASIS infrastructure.

The server implementation itself could be located at any service provider. If the service provider may change it doesn't have any effect on service users as the forwarding rules at the OASIS site can be adapted easily.