Category: technical information
Status: DRAFT
Document: eugridpma-namespaces-format-spec-20060113-0-1-4.doc
Editor: David Groep
Last updated: Mon, 16 January 2006

Namespaces Format Specification

Abstract
This document describes the format and parsing rules for the *namespaces* file as shipped with the EUGridPMA and IGTF distributions of the accredited authorities. It augments the existing *signing_policy* scheme for relying-party defined name constraints on the valid subject identifiers from trusted identity providers.

This document describes the specific expression of this namespace constraints policy as a policy file stored in a file system, and on the processing and interpretation semantics of the policy file by compliant software implementations.

Table of Contents

1 Scope of this document

Coordination of the name space of authentication credentials is of importance, since authorization and access control mechanisms traditionally leverage the uniqueness of the subject identifier in identity tokens for granting rights and privileges. The IGTF arbitrates the namespace of subject identifiers, such that a specific identifier is assigned to one and only one identity within the entire Federation. Also relying parties, at their discretion, can apply additional or other constraints on the identifier namespace(s) they accept. Such namespace constraints are thus fundamentally different from those that are expressed in the CA certificates themselves (i.e. the NameConstraints referred to in RFC 2459). The namespaces file described in this document defines which sets or subsets of identifiers comply with the policy accepted by the relying party.

This document describes the specific expression of this namespace constraints policy as a policy file stored in a file system, and on the processing and interpretation semantics of the policy file by compliant software implementations.

2 Namespace constraints file format

The namespace constraints are expressed as files in the file system that must be stored in the same file system area that also contains the actual trust anchors (root certificates). The files must be named "*hash*.namespaces", where *hash* is the lower-case hexadecimal representation of the 64-bit compacted MD5 hash value of the DER-encoded issuer name (the openssl "c_hash"). There shall be a single namespace for each issuer.

Each *namespaces* file consists of zero or more namespace policy statements. A single policy statement defined a policy for a single issuer to allow or deny signing a single subject namespace. Each policy statement shall thus have the form

```
TO Issuer "issuerDirectoryName" {PERMIT,DENY} Subject "subjectDirectoryNameExpr"
```

- The issuer to which this policy pertains is designated by the *issuerDirectoryName*, which is a quoted string representation of the X.509 subject name of the issuer certificate. The issuer name is preceded by the tokens "TO Issuer".
- The (quoted) *issuerDirectoryName* may be replaced by the single token "SELF", in which case this policy applies to the issuer whose hash corresponds to the hash contained in the namespaces file name.
- The policy can either permit or deny the issuer the right to issue subjects with specific names. A denial overrides any permissive statements in the same file.
- The *subjectDirectoryNameExpr*, which always follows a "Subject" token, is a basic regular expression that defined the set of permissible or denied subject distinguished names. For comparison, the string representation of the subject name of the identity being validated must be constructed via the rules in section 2.1.2, and then this string representation matched against the *subjectDirectoryNameExpr*.

2.1.1 File parsing rules

Policy statements may be broken across multiple lines, provided the line continuation sequence ASCII 5C ASCII 0A (a single backslash at the end of the line) is used. Any un-escaped whitespace on the subsequent line should be ignored.

Empty lines, as well as all tokens that follow an un-escaped pound sign on the same line, are not policy statements must be ignored when parsing policy. Lines are terminated with a single linefeed character (ASCII 0A). Any un-escaped carriage-return characters (ASCII 0D) must be ignored.

Each policy file may contain a single line containing "#NAMESPACES-VERSION: 1.0<newline>" that designates the version format of the *namespaces* file. Files that do not contain this line must be interpreted according to version 1.0 of the namespaces policy format.

The tokens in the policy statement (such as "TO", "Issuer", "SELF", "DENY", or "Subject") are not case sensitive.

2.1.2 Format for directoryNames

Contrary to the definitions in RFC 2253, the string rendering of directoryNames in the namespaces file is an ordered and slash-separated concatenation of the relative distinguished names. The attribute types will be represented as strings according to RFC 2256, using short names where a short name is defined in the RFC. If neither a short or a long name is defined, the numeric OID value will be used.
If the RDNSequence is an empty sequence, the result is the empty or zero length string. Otherwise, the output consists of the string encodings of each RelativeDistinguishedName in the RDNSequence (according to 2.2), starting with the first element of the sequence and moving forward toward the last. The encodings of adjoining RelativeDistinguishedNames are separated by a slash character ('/' ASCII 2F).

2.1.2.1 Important notices

- This format has been decided in accordance with the requests of the European Middleware Security Coordination Group (MWSG). It is known to violate RFC2253. We are open to (simultaneously) producing alternative namespace files that are RFC2253 compliant. Software implementers seeking to use the namespace constraints file that aim to be RFC2253 compliance are warmly invited to contact the authors.
- Not all version of the popular OpenSSL library are consistent in making string representations of X.509 directoryNames. In particular, attribute types such as *emailAddress* and *serialNumber* are known to be represented in proprietary formats like *Email* or *SN*. Note that the latter conflicts with the standard shortname for *surname*.

3 Interpretation of the namespaces file in a collection

A namespaces file consists of policies that apply to a specific root of trust (authority) and to all subordinates of that authority for which no explicit policy collection is specified. Any specific policy expression contained in the collection applies only to the authority whose subject name matches the "TO Issuer" predicate in that specific policy expression.

If a namespaces file is absent for a specific authority, and the root of trust is self-signed, the namespace of this authority is not constraint by this method.

If a namespaces file is present for a specific authority, only those namespaces explicitly permitted in the namespace policy are allowed. Certificates issued by authorities for which a namespace policy is defined, but whose subject name is not part of the permitted namespace, are invalid and MUST be rejected.

If a namespaces file is absent for a specific authority, but this authority is a subordinate of an authority for which a policy collection is specified, the policy collection of the superior issuer MUST be applied to the subordinate authority, but only policy statements in that collection that apply to the specific authority (i.e. where the *directoryName* specified in the "TO Issuer" predicate matches the subject name of the specific authority) should be considered.

4 Non-normative examples

The SWITCH Grid CA hierarchy "*7b2d086c.namespaces*"

```
##############################################################################
#NAMESPACES-VERSION: 1.0
#
# CA alias    : SWITCH hierarchy from SwissSign-Root
#

TO Issuer \
  "/C=CH/O=SwissSign/CN=SwissSign CA (RSA IK May 6 1999 18:00:58)/emailAddress=ca@SwissSign.com" \
  PERMIT Subject "/CN=SwissSign Bronze CA/emailAddress=bronze@swisssign.com/O=SwissSign/C=CH"

TO Issuer "/CN=SwissSign Bronze CA/emailAddress=bronze@swisssign.com/O=SwissSign/C=CH" \
  PERMIT Subject "/CN=SwissSign Silver CA/emailAddress=silver@swisssign.com/O=SwissSign/C=CH"

TO Issuer "/CN=SwissSign Silver CA/emailAddress=silver@swisssign.com/O=SwissSign/C=CH" \
  PERMIT Subject \
  "/CN=SWITCH CA/emailAddress=switch.ca@switch.ch/O=Switch - Teleinformatikdienste fuer Lehre und Forschung/C=CH"

TO Issuer "/CN=SWITCH CA/emailAddress=switch.ca@switch.ch/O=Switch - Teleinformatikdienste fuer Lehre und Forschung/C=CH" \
  DENY Subject "/C=CH/O=CERN/.*"

TO Issuer "/CN=SWITCH CA/emailAddress=switch.ca@switch.ch/O=Switch - Teleinformatikdienste fuer Lehre und Forschung/C=CH" \
  PERMIT Subject "/C=CH/O=.*"
```

(note here the reversed ordering of several of the intermediate SwissSign CAs!)

The DutchGrid and NIKHEF CA, a middle-of-the-road simple CA, "*16da7552.namespaces*":

```
##############################################################################
#NAMESPACES-VERSION: 1.0
#
# @(#)16da7552.namespaces
# CA alias    : NIKHEF
#    subord_of:
#    subjectDN: /C=NL/O=NIKHEF/CN=NIKHEF medium-security certification auth
#    hash     : 16da7552
#    (generated automatically from ../carep/NIKHEF/16da7552.signing_policy)
#
TO Issuer "/C=NL/O=NIKHEF/CN=NIKHEF medium-security certification auth" \
  PERMIT Subject "/O=dutchgrid/O=users/.*"

TO Issuer "/C=NL/O=NIKHEF/CN=NIKHEF medium-security certification auth" \
  PERMIT Subject "/O=dutchgrid/O=hosts/.*"
```

The UK e-Science CA, "*01621954.namespaces*", that contains an emailAddress RDN:

```
##############################################################################
#NAMESPACES-VERSION: 1.0
#
TO Issuer "/C=UK/O=eScience/OU=Authority/CN=CA/emailAddress=ca-operator@grid-support.ac.uk" \
  PERMIT Subject "/C=UK/O=eScience/.* "
```

which could also have been expressed as:

```
##############################################################################
#NAMESPACES-VERSION: 1.0
#
TO Issuer SELF  PERMIT Subject "/C=UK/O=eScience/.* "
```

5 Security Considerations

The namespace policy is an integral part of the security and protection mechanisms of a relying party, and as such should be protected from tampering at all times. In case the namespace constraints policy is distributed to the relying party by a third party, this distribution mechanism must be secured. Once obtained by the relying party, it should be adequately protected from tampering. The namespace constraints policy file proposed is not in itself integrity protected.
Implementations should realize that, in case they are not able to completely parse all statements in the namespaces file, they may inadvertently validate subjects that should have been denied. The inability to completely parse and implement any specific policy statement is a severe condition and must be duly mentioned in the appropriate logs and error messages. The only safe recourse in case of un-parseable statements is to deny access to all credentials issued by the issuer in question.

6 Author Information

David L. Groep, NIKHEF
davidg@nikhef.nl