RPDNC Public Comments

http://forge.ogf.org/sf/go/artf6311

http://www.ogf.org/gf/docs/comment.php?id=298

Aim of the Document

- Relying Party Defined Namespace Constraints (RPDNC) are **limitations on the subject namespace** issued by X.509 certification authorities (CAs) that are **defined and enforced by** the end-point at **the relying party** side. As grid authentication based on X.509 credentials provides the subject DN as a handle that identifies the authenticated entity, the capability to ensure subject name uniqueness is of critical importance in ensuring overall integrity of the authentication system.
- This document **describes the rationale** and use cases for relying party defined name space constraints, and **lists the set of desired features** a policy language expressing such constraints should have.

- The paper doesn't seem to consider a RPDNC file's life-cycle; for example, how a CA (or other agent) creates, distributes, then later either updates or invalidates RPDNC information. This may be an issue when considering how these files are distributed.

Done in new version

- The model described in the paper doesn't seem to have any explicit reference to the agent that creates the RPDNC data. (This is likely to be the CA, but you give examples where an external agent would issue an RPDNC file.) Such a description might assist when identifying who may update an RPDNC and which RPDNC should be considered.

Done in new version

- Is there an issue of trust between the CA and the agent creating the RPDNC (if that agent lies outwith the CA)? For example, if the RPDNC for some CA authorises only a subset of the namespace then one has a potential denial-of-service. Can this trust be represented, so one has some measure of how much the CA agrees with the external agent's RPDNC data?

Done in new version

- The paper describes how multiple RPDNC files might describe some DN as "allowed", "denied" or "unknown" with some overarching "ultimate-default" policy that PDP engine might use for otherwise unknown DNs. Are there alternatives to reconciling different opinions? If so, what are the advantages of this approach over others?

Moved 4.11 to a non-normative appendix – at is was too much implementation oriented. The new 4.11 text is formulated as a requirement

- Are there other implementations of this type of policy framework (multiple "yes", "no", "maybe" decisions being reconciled) that are in use? If so, can anything be learnt from how these policy frameworks are implemented and deployed?

Moved 4.11 to a non-normative appendix – at is was too much implementation oriented. The new 4.11 text is formulated as a requirement

- Is it intended for this framework to include a description of signing policies for proxy-certificates? The rule that proxy certificates are allowed, but only for certificates where the subject is the same as the issuer (modulo a "/CN=proxy") might be describable by some RPDNC policy file from a CA with end-users acting as subordinate signing agents.

  This is already described in RFC3820, and the namespace is completely predictable. Constraints should be done using the policy extension of RFC3820

SPECIFIC COMMENTS

- Page 3; 3. In the paragraph starting "We define a Relying Party Defined Namespace Constraint (RPDNC)", perhaps you should also mention here that a RPDNC file may have a default policy.

  Page 3 footnote 4: the example is rather confusing because one must distinguish between commas that separates the three examples and commas within the examples. This is compounded by the use of DC as the attributeType for the three RDNs in the DN. I would recommend stating the number of examples give and removing the comma before the "and"; for example,

  Using the LDAP representation (see RFC2253 section 2.1), the following three DNs are all initial sequences of DC=host,DC=example,DC=org: DC=org, DC=example,DC=org and DC=host,DC=example,DC=org.

  Done, kept DC since that matches CBP

  Also, perhaps changing the DC=,DC=,DC= to an example with three different attributeTypes (e.g., C, O, OU).

  Top of page 4; (section describing an engine that provides a PDP).

  I believe a PDP engine must return either "allow" or "deny" for a given DN: what should it return for "unknown"? (This isn't described here, but there is some mentioned later on.) For example, are all engine implementations required return deny (i.e., they MUST return "deny")? SHOULD they return "deny" or can they return either (e.g., subject to some system configuration or engine design)?

  Dropped text,since it was implementation specific, cf slide 6

  The paragraph describes the engine a processing exactly one policy file. I guess this is a mistake; but, if not, how is a ultimate-default-deny policy enforced if the individual PDP engines return "allow" for a DN that is "unknown". If the individual PDP engines return "deny" then they cannot support at least one of the use-cases you describe.

  If an engine may process multiple policy files then the description should be adjusted. Also, how the results from these policy files combine should also be described. For example: what if a policy file has "deny" and another has "allow" for the same DN, which one wins? Is the result "deny", "allow",
  policy-file-order-specific (e.g., first non-unknown wins in an ordered list) or engine-implementation specific? Later on there is a suggestion that order matters, but it isn't clear exactly how.

- Page 4; 4.2. d. The second sentence is confusing and could be better phrased.
  This section also introduces the concept of ordering policy files (stating it is "significant") without defining how this is so.

  Page 4; 4.5. Please bear in mind that RFC 4514 doesn't define canonical string representations. Limiting RDN attribute values to those with a single value reduces the complexity of dealing with AVAs, there are still issues with poorly defined attribute type strings (e.g., "E=" vs "Email=" vs "emailAddress="). This issue seems to be left unaddressed.

  Added footnote for this

  Page 4; footnote 5. This is the first use of the term "EE" without it being define or a reference given.

  Page 5; 4.7. With "The policy expression must support [...] at least a match-all wildcard", does "match-all wildcard" mean that any RDN must be accepted or (the more restrictive) that all RDN value of a specified RDN attributeType must match? An example of the latter (which might be written "CN=*") would match any value of CN, but not match an OU value.

  Added the wild-card works on the string rep

  Does "branch exclusions" mean specifying that no branches of any type are allowed, or that a specific named (set of) branches are explicitly forbidden?

  Edited completely

  Page 5; 4.11 Remove "not" in first sentence. (this looks like a sentence that was half-edited :-)

  The first paragraph uses terms "default-deny [policy]", "default-unknown [policy]" and "ultimate deny [policy]" without these terms having been defined. I find the paragraph is generally poorly worded. It seems to include both application behaviour with configurational advice, which is confusing. I recommend rephrasing this paragraph.

  Page 5; 5. Typo with the word "singlegle"

  Page 5; 6. It might add clarity to state what the lines mean in the diagrams. For example, "lines indicate a directed hasSignedPublicKeyOf relationship and a bidirectional hasCommonManagement relationship" (although I'm sure you guys can express this better).

- Page 5; footnote 7. This is a poor definition of "vanishingly small". It doesn't define the term and is also incorrect: it should say
  rather that "hash names MAY be used".

  Further questions: the footnote uses the term "names" without defining what a "name" is: (DNs in ASN.1, lists of RDNs?, DNs as a string); what hashing algorithms are appropriate; how many bits is sufficient (assuming a well behaved hashing algorithm)?

  Page 6; 6.1. This is the first use of term "EE" in body text; no prior definition or reference is given.

  Page 6; 6.2. First use of term "RP", but no definition or reference is given.

  Page 6; footnote 8. Could you be a little more specific than "some versions"?

  Page 7; 6.8. Many abbreviations are used without definition or a reference. Please check that terms are properly defined or referenced.

  This section seems to be where the concept of default-unknown policies are first explained; albeit in not a terribly clear fashion. I would move responsibility for defining a default-unknown policy to earlier in the document (e.g., section 3) so the terms may be used in section 4.

  This paragraph seems to introduce a new typographic convention by making "unknown" sloped/italic. It is a good idea to do this, but is must be consistent throughout the document.

Moved the implementation to the appendix already, rest of terms defined or replaced

- Page 7; 7. First para. you may also like to include "denial of service attacks" within the list of possible security issues.

  Second para. the paper has "In case the namespace constraints policy is distributed to the relying party by a third party, distribution mechanism must be secured." In fact, this requirement is present for all times, whether by relying party or not.

  "Secured" is an overly vague term. Does this include preventing eavesdropping? Identifying the remote agent supplying the RPDNC file(s)? Prevent distribution of modified versions (e.g., via man-in-the-middle attacks)? Being able to verify the authenticity of the files?

  How is this secured distribute to be achieved in practice? It seems we have a chicken-and-egg problem here: we might use TLS and/or X509-based file signing of RPDNC files, but these would require first establishing some chain of trust, requiring trusted RPDNC files.

  For comparison: root CAs are often delivered without any protection as part of an initial installation process. Once installed, these CA certificates may be used for establishing trust for subsequent updates (e.g., Debian distro signing keys).

  I feel it's important that best-practice is established. If this document doesn't establish how the secure delivery of RPDNC files is to be achieved then I feel it should have some statements describing and (as much as possible) enumerating the problems associated with establishing a secure delivery of these files.

  Addressed by new or miodified text