     Generic Naming Attributes for the Generic Security Services Application
                  Programming Interface (GSS-API)
           draft-williams-kitten-generic-naming-attributes-01

Abstract

   This document specifies several useful generic naming attributes for
   use with the Generic Security Services Application Programming
   Interface (GSS-API) Naming Extensions specified in RFC6680.

   These attributes allow applications to extract discrete components of
   a GSS-API "mechanism name" (MN) object: issuer (e.g., realm name,
   domain name, certification authority name), service and host names
   (for host-based service names), user names, and others.

Status of this Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on February 13, 2014.

Copyright Notice

Table of Contents

[1](#). Introduction and Motivation

The Generic Security Services Application Programming Interface (GSS-API) [RFC2743] allows applications -and application protocol specifications- to use various security mechanisms in a generic way. There are some shortcomings of this API that preclude a fully-generic treatment of security mechanisms. This document builds on the naming extensions to the GSS-API [RFC6680] to correct some of those shortcomings.

In [RFC6680](#) we introduced an interface by which to access "attributes" of names, but we did not specify any attributes. This document specifies some such attributes. Some of the new attributes are specifically intended to make it possible to use the GSS-API in a mechanism-generic way in common use cases where it is otherwise not possible to do so.

For example, some applications need to be able to observe the discrete elements of a peer principal's host-based service name, but they generally could only do so by parsing mechanism-specific display syntaxes or exported name token formats. Such applications are inherently not generic: they can only function correctly when used with security mechanism whose principal naming conventions/formats the applications understand.

More generally, we use the the extended naming interface to introduce an attribute model of principal naming.

[1.1](#). Naming Constraints

This document also introduces a notion of naming constraints, not unlike PKIX's [RFC5280]. Naming constraints apply to "issuers" of principal names and/or their attributes. For example, to Kerberos [RFC4120] realms, to PKIX certification authorities, to identity providers (IdPs), and so on. The goal is allow specification of policies which constrain the set of principal names that a given issuer can issue credentials for.

For example, the Kerberos realm FOO.EXAMPLE would generally not be expected to issue credentials to host-based principals in domains other than "foo.example".

For each concrete attribute specified below there are several ways to inquire a NAME's value for that attribute:

1. with naming constraint checking, providing no output if naming constraints are violated;

2. with naming constraint checking, providing an output indicator of
    naming constraint violations;

3. without naming constraint checking;

4. any of the above with "fast" (no slow I/O involved) naming
    constraint checking.

(1) is the default behavior. The others are obtained by adding an
appropriate prefix to the attribute name.

Existing security mechanisms may not have any formal notion of naming
constraints, but it is common to have some naming constraint
conventions nonetheless. For example, Kerberos realm naming
conventions are that realm names should mirror Domain Name System
(DNS) [RFC1035] domain names, and that hostnames embedded in Kerberos
principal names should a) be fully-qualified, b) within the domain
corresponding to the DNS domain name derived from the realm's name.
Or a Kerberos implementation might lookup a host's realm and check
that it matches the principal's realm. Naming constraints should be
formalized for all GSS-API security mechanisms.

1.2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in [RFC2119].

2. Generic Attributes

We add a number of generic name attributes, to be used via the GSS-
API extended naming facility [RFC6680]. Some of these attributes can
be used as prefixes of other attributes, that is, they can be used to
modify the semantics of other attributes (see section 6 of RFC6680).

Note: in all cases the display form of each attribute SHALL consist
of text using the character set, codeset, and encoding from the
caller's locale.

2.1. Concrete Attributes

These attributes generally have a single value each. Only one of
these attributes can also be used a prefix: the issuer name
attribute.

2.1.1. Issuer Name

We add an attribute by which to obtain a name of an issuer of a
mechanism name (MN) or of an attribute of an MN. The API name for
this attribute is GSS_C_ATTR_GENERIC_ISSUERNAME, and it's actual
attribute name is "urn:ietf:id:ietf-kitten-name-attrs-00-issuername".

The display form of issuer names is mechanism-specific.

The non-display form of issuer names SHALL be the exported name token
form of the issuer's name. Not all mechanisms will support issuer
names as MNs, therefore implementations MAY output a null non-display
value.

For example, for the Kerberos mechanism [RFC4121] an issuer name
would generally (but not always!) be a Kerberos realm name, probably
displayed as just the realm name. (But note that there is not yet a
Kerberos realm name as MN specification. We will specify one
separately.)

This attribute can be used as prefix of other attributes. When used
as a prefix, this attribute indicates that the application wishes to
know the name of the issuer of the prefixed attribute of the given
MN.

2.1.2. User Name

We add an attribute by which to obtain the component of an MN naming
a user. The API name for this attribute is
GSS_C_ATTR_GENERIC_USERNAME, and it's actual attribute name is
"urn:ietf:id:ietf-kitten-name-attrs-00-username".

The display form of user names is mechanism-specific.

The non-display form of user names is mechanism-specific.

2.1.3. Service Name

We add an attribute by which to obtain the component or an MN naming
a service as part of a host- or domain-based service name. The API
name for this attribute is GSS_C_ATTR_GENERIC_SERVICENAME, and it's
actual attribute name is
"urn:ietf:id:ietf-kitten-name-attrs-00-servicename".

The non-display form of the service name SHALL be the UTF-8 encoding
of the service name.

2.1.4. Host Name

We add an attribute by which to obtain the component of an MN naming
a host as part of a host- or domain-based service name. The API name
for this attribute is GSS_C_ATTR_GENERIC_HOSTNAME, and it's actual
attribute name is "urn:ietf:id:ietf-kitten-name-attrs-00-hostname".

The display form of a host name MAY be stylized and SHOULD NOT be
A-labels. [RFC5890].

The non-display form of host names SHOULD be a character string as
described in [RFC1123], and SHOULD NOT be U-labels [RFC5890].

2.1.5. Domain Name

We add an attribute by which to obtain the component of an MN naming
a domain as part of a domain-based service name. The API name for
this attribute is GSS_C_ATTR_GENERIC_DOMAINNAME, and it's actual
attribute name is "urn:ietf:id:ietf-kitten-name-attrs-00-domainname".

The display form of a domain name MAY be stylized and SHOULD NOT be
A-labels. [RFC5890].

The non-display form of domain names SHOULD be a character string as
described in [RFC1123], and SHOULD NOT be U-labels [RFC5890].

2.2. Prefix Attributes

GSS_Get_name_attribute() using attributes described in the preceding
section SHALL fail if there are any name constraints that can be
applied to the issuers of those names and, in applying those
constraints, it is discovered that the issuer was not permitted to
issue credentials for the MN.

For example, a Kerberos realm named "FOO.EXAMPLE" might not be
expected to issue credentials (tickets, keys) to host-based service
names for hosts not ending in ".foo.example" or which are not
"foo.example".

Several generic attribute prefixes are described below for overriding
this behavior.

2.2.1. GSS_C_ATTR_GENERIC_UNCONSTRAINED

This attribute prefix, named GSS_C_ATTR_GENERIC_UNCONSTRAINED in the
API, and with an actual name of
"urn:ietf:id:ietf-kitten-name-attrs-00-gen-unconstrained", indicates
that the application wants the value of the prefixed attribute
without any name constraint checking.

2.2.2. GSS_C_ATTR_GENERIC_UNCONSTRAINED_OK

This attribute prefix, named GSS_C_ATTR_GENERIC_UNCONSTRAINED_OK in
the API, and with an actual name of
"urn:ietf:id:ietf-kitten-name-attrs-00-gen-unconstrained-ok",
indicates that the application wants the value of the prefixed
attribute regardless of any applicable naming constraints, but to
indicate the name constraint status via the 'authenticated' output
parameter of the GSS_Get_name_attribute() interface.

2.2.3. GSS_C_ATTR_GENERIC_FAST

This attribute prefix, named GSS_C_ATTR_GENERIC_FAST in the API, and
with an actual name of
"urn:ietf:id:ietf-kitten-name-attrs-00-gen-fast", indicates that the
application requires that the mechanism not perform any slow
operations (e.g., connecting to a directory for the purposes of name
constraint validation) in obtaining the prefixed attribute of the
given MN.

3. Local Name Attributes

   Normally an Internet specification would not be expected to specify
   any local name attributes of GSS names. However, there is one common
   and very useful local name attribute, which we specify below.
   Implementations are free to use different names for this attribute or
   exclude it altogether -- it is a local name attribute, after all.

3.1. GSS_C_ATTR_LOCAL_LOGIN_USER

   This attribute, with suggested API symbolic name
   GSS_C_ATTR_LOCAL_LOGIN_USER, and suggested actual name "local-login-
   user", requests a local user name corresponding to the given MN, if
   any.

   Obtaining the local user name corresponding to an MN may require
   complex name mapping or lookup operations that are completely
   implementation-defined.

4. **Suggested Mechanism-Specific Name Attributes (INFORMATIONAL)**

   [[anchor1: This section should really be split out into separate
   Internet-Drafts. It is here only because the author lacks the time
   at the moment of writing to create such separate I-Ds.]]

4.1. **Suggested Kerberos-Specific Name Attributes**

   o realm (corresponding to issuer name)

   o component 0 (first component of a principal name)

   o component 1 (second component of a principal name)

   o ..

   o component 9 (10th component of a principal name)

   o components (ordered set of components of a principal name)

   o transit path (ordered set of realm and CA names)

   o specific authorization data elements

   o PKINIT client certificate

   o session key enctype

   o enctypes involved in transit path (this would only be available to
      initiators)

4.2. **Suggested PKU2U-Specific Name Attributes**

   [[anchor2: Add reference to PKU2U.]]

   o issuer CA name

   o trust path to a trust anchor

   o certificate

   o certificate subject public key

   o certificate subject name

   o certificate subject alternate names

o specific certificate extensions

o certificate algorithm names

o session key enctype

5. Security Considerations

[Add text regarding name constraint checking and explaining the
default-to-safe design of the generic name attributes defined in
section 2.]

6. IANA Considerations

   [Add text regarding the registration and assignment of the name
   attributes described in the preceding sections. In particular we
   should want these attributes' names to not reflect an Internet-Draft
   name, but an RFC number.]

7. References

7.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC1035] Mockapetris, P., "Domain names - implementation and
              specification", STD 13, RFC 1035, November 1987.

   [RFC1123] Braden, R., "Requirements for Internet Hosts - Application
              and Support", STD 3, RFC 1123, October 1989.

   [RFC2743] Linn, J., "Generic Security Service Application Program
              Interface Version 2, Update 1", RFC 2743, January 2000.

   [RFC2744] Wray, J., "Generic Security Service API Version 2 :
              C-bindings", RFC 2744, January 2000.

   [RFC5890] Klensin, J., "Internationalized Domain Names for
              Applications (IDNA): Definitions and Document Framework",
              RFC 5890, August 2010.

   [RFC6680] Williams, N., Johansson, L., Hartman, S., and S.
              Josefsson, "Generic Security Service Application
              Programming Interface (GSS-API) Naming Extensions",
              RFC 6680, August 2012.

7.2. Informative References

   [RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The
              Kerberos Network Authentication Service (V5)", RFC 4120,
              July 2005.

   [RFC4121] Zhu, L., Jaganathan, K., and S. Hartman, "The Kerberos
              Version 5 Generic Security Service Application Program
              Interface (GSS-API) Mechanism: Version 2", RFC 4121,
              July 2005.

   [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S.,
              Housley, R., and W. Polk, "Internet X.509 Public Key
              Infrastructure Certificate and Certificate Revocation List
              (CRL) Profile", RFC 5280, May 2008.

Author's Address

   Nicolas Williams
   Cryptonector, LLC

   Email: nico@cryptonector.com