

NETWORK WORKING GROUP
Internet-Draft
Intended status: Standards Track
Expires: October 3, 2009

N. Williams
Sun
April 1, 2009

Extended Generic Security Service Mechanism Inquiry APIs
draft-ietf-kitten-extended-mech-inquiry-06.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on October 3, 2009.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This document introduces new application programming interfaces (APIs) to the Generic Security Services API (GSS-API) for extended mechanism attribute inquiry. These interfaces are primarily intended

to reduce instances of hardcoding of mechanism identifiers in GSS applications.

These interfaces include: mechanism attributes and attribute sets, a function for inquiring the attributes of a mechanism, a function for indicating mechanisms that posses given attributes, and a function for displaying mechanism attributes.

Table of Contents

1.	Conventions used in this document	3
2.	Introduction	3
3.	New GSS-API Interfaces	3
3.1.	Mechanism Attributes and Attribute Sets	3
3.2.	List of Known Mechanism Attributes	4
3.3.	Mechanism Attribute Sets of Existing Mechs	6
3.4.	New GSS-API Function Interfaces	7
3.4.1.	Mechanism Attribute Criticality	8
3.4.2.	GSS_Indicate_mechs_by_attr()	8
3.4.3.	GSS_Inquire_attrs_for_mech()	9
3.4.4.	GSS_Display_mech_attr()	9
3.4.5.	New Major Status Values	10
3.4.6.	C-Bindings	10
4.	Requirements for Mechanism Designers	11
5.	IANA Considerations	12
6.	Security considerations	12
7.	References	12
7.1.	Normative References	12
7.2.	Informative References	12
	Author's Address	13

1. 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. Introduction

GSS-API [\[RFC2743\]](#) mechanisms have a number of properties that may be of interest to applications. The lack of APIs for inquiring about available mechanisms' properties has meant that many GSS-API applications must hardcode mechanism OIDs. Ongoing work may result in a variety of new GSS-API mechanisms. Applications should not have to hardcode their OIDs.

For example, the SSHv2 protocol [\[RFC4251\]](#) supports the use of GSS-API mechanisms for authentication [\[RFC4462\]](#), but it explicitly prohibits the use of SPNEGO [\[RFC4178\]](#). Future mechanisms that negotiate mechanisms would have to be forbidden as well, but there is no way to implement applications that inquire what mechanisms are available and then programmatically exclude mechanisms "like SPNEGO".

3. New GSS-API Interfaces

We introduce a new concept: that of mechanism attributes. By allowing applications to query the set of attributes associated with individual mechanisms and to find out which mechanisms support a given set of attributes we allow applications to select mechanisms based on their attributes yet without having to hardcode mechanism OIDs.

[Section 3.1](#) describes the mechanism attributes concept. Sections 3.4.2, 3.4.3 and 3.4.4 describe three new interfaces that deal in mechanisms and attribute sets:

- o GSS_Indicate_mechs_by_attrs()
- o GSS_Inquire_attrs_for_mech()
- o GSS_Display_mech_attr()

3.1. Mechanism Attributes and Attribute Sets

An abstraction for the features provided by mechanisms and pseudo-mechanisms is needed in order to facilitate the programmatic selection of mechanisms. Pseudo-mechanisms are mechanisms which make reference to other mechanisms in order to provide their services. For example, SPNEGO is a pseudo-mechanism, for without other

mechanisms SPNEGO is useless.

Two data types are needed: one for individual mechanism attributes and one for mechanism attribute sets. To simplify the mechanism attributes interfaces we reuse the 'OID' and 'OID set' data types and model individual mechanism attribute types as OIDs.

To this end we define an open namespace of mechanism attributes and assign them arcs off of this OID:

<TBA by IANA> [1.3.6.1.5.5.13 appears to be available; see <http://www.iana.org/assignments/smi-numbers>]

Each mechanism has a set of mechanism attributes that it supports as described in its specification.

3.2. List of Known Mechanism Attributes

Mech Attr Name	OID Arc	Arc Name
GSS_C_MA_MECH_CONCRETE	(1)	concrete-mech
GSS_C_MA_MECH_PSEUDO	(2)	pseudo-mech
GSS_C_MA_MECH_COMPOSITE	(3)	composite-mech
GSS_C_MA_MECH_NEGO	(4)	mech-negotiation-mech
GSS_C_MA_MECH_GLUE	(5)	mech-glue
GSS_C_MA_NOT_MECH	(6)	not-mech
GSS_C_MA_DEPRECATED	(7)	mech-deprecated
GSS_C_MA_NOT_DFLT_MECH	(8)	mech-not-default
GSS_C_MA_ITOK_FRAMED	(9)	initial-is-framed
GSS_C_MA_AUTH_INIT	(10)	auth-init-princ
GSS_C_MA_AUTH_TARG	(11)	auth-targ-princ
GSS_C_MA_AUTH_INIT_INIT	(12)	auth-init-princ-initial
GSS_C_MA_AUTH_TARG_INIT	(13)	auth-targ-princ-initial
GSS_C_MA_AUTH_INIT_ANON	(14)	auth-init-princ-anon
GSS_C_MA_AUTH_TARG_ANON	(15)	auth-targ-princ-anon
GSS_C_MA_DELEG_CRED	(16)	deleg-cred
GSS_C_MA_INTEG_PROT	(17)	integ-prot
GSS_C_MA_CONF_PROT	(18)	conf-prot
GSS_C_MA_MIC	(19)	mic
GSS_C_MA_WRAP	(20)	wrap
GSS_C_MA_PROT_READY	(21)	prot-ready
GSS_C_MA_REPLAY_DET	(22)	replay-detection
GSS_C_MA_OOS_DET	(23)	oos-detection
GSS_C_MA_CBINDINGS	(24)	channel-bindings
GSS_C_MA_PFS	(25)	pfs
GSS_C_MA_COMPRESS	(26)	compress
GSS_C_MA_CTX_TRANS	(27)	context-transfer

Williams

Expires October 3, 2009

[Page 4]

```

| <reserved>                | (28..) |
+-----+-----+-----+

```

Table 1

Mech Attr Name	Purpose
GSS_C_MA_MECH_CONCRETE	Indicates that a mech is neither a pseudo- mechanism nor a composite mechanism.
GSS_C_MA_MECH_PSEUDO	Indicates that a mech is a pseudo-mechanism.
GSS_C_MA_MECH_COMPOSITE	Indicates that a mech is a composite of other mechanisms. This is reserved for a specification of "stackable" pseudo-mechanisms.
GSS_C_MA_MECH_NEGO	Indicates that a mech negotiates other mechs (e.g., SPNEGO has this attribute).
GSS_C_MA_MECH_GLUE	Indicates that the OID is not for a mechanism but for the GSS-API itself.
GSS_C_MA_NOT_MECH	Indicates that the OID is known, yet also known not to be the OID of any GSS-API mechanism (or the GSS-API itself).
GSS_C_MA_DEPRECATED	Indicates that a mech (or its OID) is deprecated and MUST NOT be used as a default mechanism.
GSS_C_MA_NOT_DFLT_MECH	Indicates that a mech (or its OID) MUST NOT be used as a default mechanism.
GSS_C_MA_ITOK_FRAMED	Indicates that the given mechanism's initial context tokens are properly framed as per section 3.1 of rfc2743 .
GSS_C_MA_AUTH_INIT	Indicates support for authentication of initiator to acceptor.
GSS_C_MA_AUTH_TARG	Indicates support for authentication of acceptor to initiator.
GSS_C_MA_AUTH_INIT_INIT	Indicates support for "initial" authentication of initiator to acceptor. "Initial authentication" refers to the use of passwords, or keys stored on tokens, for authentication. Whether a mechanism supports initial authentication may depend on IETF consensus (see Security Considerations).
GSS_C_MA_AUTH_TARG_INIT	Indicates support for initial

	authentication of acceptor to initiator.
GSS_C_MA_AUTH_INIT_ANON	Indicates support for GSS_C_NT_ANONYMOUS as an initiator principal name.
GSS_C_MA_AUTH_TARG_ANON	Indicates support for GSS_C_NT_ANONYMOUS as a target principal name.
GSS_C_MA_DELEG_CRED	Indicates support for credential delegation.
GSS_C_MA_INTEG_PROT	Indicates support for per-message integrity protection.
GSS_C_MA_CONF_PROT	Indicates support for per-message confidentiality protection.
GSS_C_MA_MIC	Indicates support for MIC tokens.
GSS_C_MA_WRAP	Indicates support for WRAP tokens.
GSS_C_MA_PROT_READY	Indicates support for per-message protection prior to full context establishment.
GSS_C_MA_REPLAY_DET	Indicates support for replay detection.
GSS_C_MA_OOS_DET	Indicates support for out-of-sequence detection.
GSS_C_MA_CBINDINGS	Indicates support for channel bindings.
GSS_C_MA_PFS	Indicates support for Perfect Forward Security.
GSS_C_MA_COMPRESS	Indicates support for compression of data inputs to GSS_Wrap().
GSS_C_MA_CTX_TRANS	Indicates support for security context export/import.

Table 2

3.3. Mechanism Attribute Sets of Existing Mechs

The Kerberos V mechanism [[RFC1964](#)] provides the following mechanism attributes:

- ```

0 GSS_C_MA_MECH_CONCRETE
0 GSS_C_MA_ITOK_FRAMED
0 GSS_C_MA_AUTH_INIT
0 GSS_C_MA_AUTH_TARG
0 GSS_C_MA_DELEG_CRED
0 GSS_C_MA_INTEG_PROT
0 GSS_C_MA_CONF_PROT
0 GSS_C_MA_MIC
0 GSS_C_MA_WRAP
0 GSS_C_MA_PROT_READY

```

Williams

Expires October 3, 2009

[Page 6]

- o GSS\_C\_MA\_REPLAY\_DET
- o GSS\_C\_MA\_OOS\_DET
- o GSS\_C\_MA\_CBINDINGS
- o GSS\_C\_MA\_CTX\_TRANS (some implementations, using implementation-specific exported context token formats)

The Kerberos V mechanism also has a deprecated OID which has the same mechanism attributes as above, and GSS\_C\_MA\_DEPRECATED.

The mechanism attributes of the SPKM [RFC2025] family of mechanisms will be provided in a separate document as SPKM is current being reviewed for possibly significant changes due to problems in its specifications.

The LIPKEY mechanism [RFC2847] offers the following attributes:

- o GSS\_C\_MA\_MECH\_CONCRETE
- o GSS\_C\_MA\_ITOK\_FRAMED
- o GSS\_C\_MA\_AUTH\_INIT\_INIT
- o GSS\_C\_MA\_AUTH\_TARG (from SPKM-3)
- o GSS\_C\_MA\_AUTH\_TARG\_ANON (from SPKM-3)
- o GSS\_C\_MA\_INTEG\_PROT
- o GSS\_C\_MA\_CONF\_PROT
- o GSS\_C\_MA\_REPLAY\_DET
- o GSS\_C\_MA\_OOS\_DET
- o GSS\_C\_MA\_CTX\_TRANS (some implementations, using implementation-specific exported context token formats)

(LIPKEY should also provide GSS\_C\_MA\_CBINDINGS, but SPKM-3 requires clarifications on this point.)

The SPNEGO mechanism [RFC4178] provides the following attributes:

- o GSS\_C\_MA\_MECH\_NEGO
- o GSS\_C\_MA\_ITOK\_FRAMED

All other mechanisms' attributes will be described elsewhere.

### **3.4. New GSS-API Function Interfaces**

Several new interfaces are given by which, for example, GSS-API applications may determine what features are provided by a given mechanism and what mechanisms provide what features.

These new interfaces are all OPTIONAL.

Applications should use GSS\_Indicate\_mechs\_by\_attr() instead of GSS\_Indicate\_mechs() wherever possible.



Applications can use `GSS_Indicate_mechs_by_attr()` to determine what, if any, mechanisms provide a given set of features.

`GSS_Indicate_mechs_by_attr()` can also be used to indicate (as in `GSS_Indicate_mechs()`) the set of available mechanisms of each type (concrete, mechanism negotiation pseudo-mechanism, etc.).

#### **3.4.1. Mechanism Attribute Criticality**

Mechanism attributes may be added at any time. Not only may attributes be added to the list of known mechanism attributes at any time, but the set of mechanism attributes supported by a mechanism can be changed at any time.

For example, new attributes might be added to reflect whether a mechanism's initiator must contact online infrastructure, and/or whether the acceptor must do so. In this example the Kerberos V mechanism would gain a new attribute even though the mechanism itself is not modified.

Applications making use of attributes not defined herein then would have no way of knowing whether a GSS-API implementation and its mechanisms know about new mechanism attributes. To address this problem `GSS_Indicate_mechs_by_attr()` and `GSS_Indicate_mechs_by_attr()` support a notion of critical mechanism attributes. Applications can search for mechanisms that understand mechanism attributes that are critical to the application, and the application may ask what mechanism attributes are understood by a given mechanism.

#### **3.4.2. `GSS_Indicate_mechs_by_attr()`**

Inputs:

- o `desired_mech_attrs` SET OF OBJECT IDENTIFIER -- set of `GSS_C_MA_*` OIDs that the mechanisms indicated in the `mechs` output parameter MUST offer
- o `except_mech_attrs` SET OF OBJECT IDENTIFIER -- set of `GSS_C_MA_*` OIDs that the mechanisms indicated in the `mechs` output parameter MUST NOT offer
- o `critical_mech_attrs` SET OF OBJECT IDENTIFIER -- set of `GSS_C_MA_*` OIDs that the mechanisms indicated in the `mechs` output parameter MUST understand (i.e., `mechs` must know whether critical attributes are or are not supported)

Outputs:

- o `major_status` INTEGER
- o `minor_status` INTEGER
- o `mechs` SET OF OBJECT IDENTIFIER -- set of mechanisms that support the given `desired_mech_attrs` but not the `except_mech_attrs`, and



all of which understand the given `critical_mech_attrs` (the caller must release this output with `GSS_Release_oid_set()`)

Return `major_status` codes:

- o `GSS_S_COMPLETE` indicates success; the output `mechs` parameter MAY be the empty set (`GSS_C_NO_OID_SET`).
- o `GSS_S_FAILURE` indicates that the request failed for some other reason.

`GSS_Indicate_mechs_by_attrs()` returns the set of OIDs corresponding to mechanisms that offer at least the desired `mech_attrs` but none of the except `mech_attrs`, and which understand all of the attributes listed in `critical_mech_attrs`.

When all three set of OID input parameters are the empty set this function acts as a version of `GSS_indicate_mechs()` that outputs the set of all supported mechanisms.

#### **3.4.3. GSS\_Inquire\_attrs\_for\_mech()**

Inputs:

- o `mech` OBJECT IDENTIFIER -- mechanism OID

Outputs:

- o `major_status` INTEGER
- o `minor_status` INTEGER
- o `mech_attrs` SET OF OBJECT IDENTIFIER -- set of `mech_attrs` OIDs (`GSS_C_MA_*`) supported by the mechanism (the caller must release this output with `GSS_Release_oid_set()`)
- o `known_mech_attrs` SET OF OBJECT IDENTIFIER -- set of `mech_attrs` OIDs known to the mechanism implementation (the caller must release this output with `GSS_Release_oid_set()`).

Return `major_status` codes:

- o `GSS_S_COMPLETE` indicates success; the output `mech_attrs` parameter MAY be the empty set (`GSS_C_NO_OID_SET`).
- o `GSS_S_BAD_MECH` indicates that the mechanism named by the `mech` parameter does not exist or that `mech` is `GSS_C_NO_OID` and no default mechanism could be determined.
- o `GSS_S_FAILURE` indicates that the request failed for some other reason.

`GSS_Inquire_mech_attrs_for_mech()` indicates the set of mechanism attributes supported by a given mechanism.

#### **3.4.4. GSS\_Display\_mech\_attr()**

Inputs:





- o mech\_attr OBJECT IDENTIFIER -- mechanism attribute OID

Outputs:

- o major\_status INTEGER
- o minor\_status INTEGER
- o name OCTET STRING, -- name of mechanism attribute (e.g., GSS\_C\_MA\_\*)
- o short\_desc OCTET STRING, -- a short description of the mechanism attribute (the caller must release this output with GSS\_Release\_buffer())
- o long\_desc OCTET STRING -- a longer description of the mechanism attribute (the caller must release this output with GSS\_Release\_buffer())

Return major\_status codes:

- o GSS\_S\_COMPLETE indicates success.
- o GSS\_S\_BAD\_MECH\_ATTR indicates that the mechanism attribute referenced by the mech\_attr parameter is unknown to the implementation.
- o GSS\_S\_FAILURE indicates that the request failed for some other reason.

This function can be used to obtain human-readable descriptions of GSS-API mechanism attributes.

#### **3.4.5. New Major Status Values**

A single new major status code is added for GSS\_Display\_mech\_attr():

- o GSS\_S\_BAD\_MECH\_ATTR  
roughly corresponding to GSS\_S\_BAD\_MECH, but applicable to mechanism attribute OIDs, rather than to mechanism OIDs.

For the C-bindings of the GSS-API [[RFC2744](#)] GSS\_S\_BAD\_MECH\_ATTR shall have a routine error number of 19 (this is shifted to the left by GSS\_C\_ROUTINE\_ERROR\_OFFSET).

#### **3.4.6. C-Bindings**

Note that there is a bug in the C bindings of the GSS-APIv2u1 [[RFC2744](#)] in that the C 'const' attribute is applied to types which are pointer typedefs. This is a bug because this declares that the pointer argument is 'const' rather than that the object pointed by it is const. To avoid this error we hereby define new typedefs which include const properly:

```
typedef const gss_buffer_desc * gss_const_buffer_t;
typedef const struct gss_channel_bindings_struct *
 gss_const_channel_bindings_t;
```



```
typedef const <platform-specific> gss_const_ctx_id_t;
typedef const <platform-specific> gss_const_cred_id_t;
typedef const <platform-specific> gss_const_name_t;
typedef const gss_OID_desc * gss_const_OID;
typedef const gss_OID_set_desc * gss_const_OID_set;
```

Figure 1: const typedefs

Note that only `gss_const_OID` and `gss_const_OID_set` are used below. We include the other const typedefs for convenience since the C bindings of the GSS-API do use const with pointer typedefs when it should often instead use the above typedefs instead.

```
#define GSS_S_BAD_MECH_ATTR (19ul << GSS_C_ROUTINE_ERROR_OFFSET)

OM_uint32 gss_inquire_mechs_for_attrs(
 OM_uint32 *minor_status,
 gss_const_OID_set desired_mech_attrs,
 gss_const_OID_set except_mech_attrs,
 gss_const_OID_set critical_mech_attrs,
 gss_OID_set *mechs);

OM_uint32 gss_inquire_attrs_for_mech(
 OM_uint32 *minor_status,
 gss_const_OID mech,
 gss_OID_set *mech_attrs,
 gss_OID_set *known_mech_attrs);

OM_uint32 gss_display_mech_attr(
 OM_uint32 *minor_status,
 gss_const_OID mech_attr,
 gss_buffer_t name,
 gss_buffer_t short_desc,
 gss_buffer_t long_desc);
```

Figure 2: C bindings

Note that output buffers must be released via `gss_release_buffer()`. Output OID sets must be released via `gss_release_oid_set()`.

#### **4. Requirements for Mechanism Designers**

All future GSS-API mechanism specifications MUST:

- o list the set of GSS-API mechanism attributes associated with them



## **5. IANA Considerations**

The namespace of programming language symbols with names beginning with GSS\_C\_MA\_\* is reserved for allocation by IESG Protocol Action. The IANA should allocate a base OID, as an arc of 1.3.6.1.5.5, for the set of GSS\_C\_MA\_\* described herein, and it should register all of the GSS\_C\_MA\_\* values described in [Section 3.2](#)

## **6. Security considerations**

This document specifies extensions to a security-related API. It imposes new requirements on future GSS-API mechanisms, and the specification of future protocols that use the GSS-API should make reference to this document where applicable. The ability to inquire about specific properties of mechanisms should improve security.

The semantics of each mechanism attribute may include a security component.

Application developers must understand that mechanism attributes may be added at any time, both, to the set of known mechanism attributes, as well as to existing mechanism's sets of supported mechanism attributes. Therefore application developers using the APIs described herein must understand what mechanism attributes their applications depend critically on, and must use the mechanism attribute criticality features of these APIs.

## **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.
- [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.

### **7.2. Informative References**

- [RFC1964] Linn, J., "The Kerberos Version 5 GSS-API Mechanism", [RFC 1964](#), June 1996.
- [RFC2025] Adams, C., "The Simple Public-Key GSS-API Mechanism"



(SPKM)", [RFC 2025](#), October 1996.

- [RFC2847] Eisler, M., "LIPKEY - A Low Infrastructure Public Key Mechanism Using SPKM", [RFC 2847](#), June 2000.
- [RFC4178] Zhu, L., Leach, P., Jaganathan, K., and W. Ingersoll, "The Simple and Protected Generic Security Service Application Program Interface (GSS-API) Negotiation Mechanism", [RFC 4178](#), October 2005.
- [RFC4251] Ylonen, T. and C. Lonvick, "The Secure Shell (SSH) Protocol Architecture", [RFC 4251](#), January 2006.
- [RFC4462] Hutzelman, J., Salowey, J., Galbraith, J., and V. Welch, "Generic Security Service Application Program Interface (GSS-API) Authentication and Key Exchange for the Secure Shell (SSH) Protocol", [RFC 4462](#), May 2006.

#### Author's Address

Nicolas Williams  
Sun Microsystems  
5300 Riata Trace Ct  
Austin, TX 78727  
US

Email: [Nicolas.Williams@sun.com](mailto:Nicolas.Williams@sun.com)



