NAME

gss_add_cred - Construct credentials incrementally

SYNOPSIS

#include <gssapi/gssapi.h>

OM_uint32
gss_add_cred(*OM_uint32 *minor_status, const gss_cred_id_t input_cred_handle, const gss_name_t desired_name, const gss_OID desired_mech, gss_cred_usage_t cred_usage, OM_uint32 initiator_time_req, OM_uint32 acceptor_time_req, gss_cred_id_t *output_cred_handle, gss_OID_set *actual_mechs, OM_uint32 *initiator_time_rec, OM_uint32 *acceptor_time_rec*);

DESCRIPTION

Adds a credential-element to a credential. The credential-element is identified by the name of the principal to which it refers. GSS-API implementations must impose a local access-control policy on callers of this routine to prevent unauthorized callers from acquiring credential-elements to which they are not entitled. This routine is not intended to provide a "login to the network" function, as such a function would involve the creation of new mechanism-specific authentication data, rather than merely acquiring a GSS-API handle to existing data. Such functions, if required, should be defined in implementation-specific extensions to the API.

If *desired_name* is GSS_C_NO_NAME, the call is interpreted as a request to add a credential element that will invoke default behavior when passed to **gss_init_sec_context**() (if cred_usage is GSS_C_INITIATE or GSS_C_BOTH) or **gss_accept_sec_context**() (if *cred_usage* is GSS_C_ACCEPT or GSS_C_BOTH).

This routine is expected to be used primarily by context acceptors, since implementations are likely to provide mechanism-specific ways of obtaining GSS-API initiator credentials from the system login process. Some implementations may therefore not support the acquisition of GSS_C_INITIATE or GSS_C_BOTH credentials via **gss_acquire_cred**() for any name other than GSS_C_NO_NAME, or a name produced by applying either **gss_inquire_cred**() to a valid credential, or **gss_inquire_context**() to an active context.

If credential acquisition is time-consuming for a mechanism, the mechanism may choose to delay the actual acquisition until the credential is required (e.g. by **gss_init_sec_context**() or **gss_accept_sec_context**(*)*.) Such mechanism-specific implementation decisions should be invisible to the calling application; thus a call of **gss_inquire_cred**() immediately following the call of **gss_add_cred**() must return valid credential data, and may therefore incur the overhead of a deferred credential acquisition.

This routine can be used to either compose a new credential containing all credential-elements of the original in addition to the newly-acquire credential-element, or to add the new credential-element to an existing credential. If NULL is specified for the *output_cred_handle* parameter argument, the new credential-element will be added to the credential identified by *input_cred_handle*; if a valid pointer is specified for the *output_cred_handle* parameter, a new credential handle will be created.

If GSS_C_NO_CREDENTIAL is specified as the *input_cred_handle*, **gss_add_cred**() will compose a credential (and set the *output_cred_handle* parameter accordingly) based on default behavior. That is, the call will have the same effect as if the application had first made a call to **gss_acquire_cred**(), specifying the same usage and passing GSS_C_NO_NAME as the *desired_name* parameter to obtain an explicit credential handle embodying default behavior, passed this credential handle to **gss_add_cred**(), and finally called **gss_release_cred**() on the first credential handle.

If GSS_C_NO_CREDENTIAL is specified as the *input_cred_handle* parameter, a non- NULL *output_cred_handle* must be supplied.

PARAMETERS

minor_status — Mechanism specific status code.

input_cred_handle — The credential to which a credential-element will be added. If GSS_C_NO_CREDENTIAL is specified, the routine will compose the new credential based on default behavior (see description above). Note that, while the credential-handle is not modified by **gss_add_cred**(), the underlying credential will be modified if *output_credential_handle* is NULL.

desired_name — Name of principal whose credential should be acquired.

desired_mech — Underlying security mechanism with which the credential may be used.

cred_usage

- GSS_C_BOTH — Credential may be used either to initiate or accept security contexts.
- GSS_C_INITIATE — Credential will only be used to initiate security contexts.
- GSS_C_ACCEPT — Credential will only be used to accept security contexts.

initiator_time_req — Number of seconds that the credential should remain valid for initiating security contexts. This argument is ignored if the composed credentials are of type

GSS_C_ACCEPT. Specify GSS_C_INDEFINITE to request that the credentials have the maximum permitted initiator lifetime.

acceptor_time_req — Number of seconds that the credential should remain valid for accepting security contexts. This argument is ignored if the composed credentials are of type GSS_C_INITIATE. Specify GSS_C_INDEFINITE to request that the credentials have the maximum permitted initiator lifetime.

output_cred_handle — The returned credential handle, containing the new credential-element and all the credential-elements from *input_cred_handle*. If a valid pointer to a *gss_cred_id_t* is supplied for this parameter, **gss_add_cred**() creates a new credential handle containing all credential-elements from the *input_cred_handle* and the newly acquired credential-element; if NULL is specified for this parameter, the newly acquired credential-element will be added to the credential identified by *input_cred_handle*.

The resources associated with any credential handle returned via this parameter must be released by the application after use with a call to **gss_release_cred**().

actual_mechs — The complete set of mechanisms for which the new credential is valid. Storage for the returned OID-set must be freed by the application after use with a call to **gss_release_oid_set**(). Specify NULL if not required.

initiator_time_rec — Actual number of seconds for which the returned credentials will remain valid for initiating contexts using the specified mechanism. If the implementation or mechanism does not support expiration of credentials, the value GSS_C_INDEFINITE will be returned. Specify NULL if not required.

acceptor_time_rec — Actual number of seconds for which the returned credentials will remain valid for accepting security contexts using the specified mechanism. If the implementation or mechanism does not support expiration of credentials, the value GSS_C_INDEFINITE will be returned. Specify NULL if not required.

RETURN VALUES

GSS_S_COMPLETE — Successful completion.

GSS_S_BAD_MECH — Unavailable mechanism requested.

GSS_S_BAD_NAMETYPE — Type contained within desired_name parameter is not supported

GSS_S_BAD_NAME	Value supplied for desired_name parameter is ill-formed.
GSS_S_DUPLICATE_ELEMENT	The credential already contains an element for the requested mechanism with overlapping usage and validity period.
GSS_S_CREDENTIALS_EXPIRED	The required credentials could not be added because they have expired.
GSS_S_NO_CRED	No credentials were found for the specified name.

SEE ALSO

gss_accept_sec_context(3), gss_acquire_cred(3), gss_init_sec_context(3), gss_inquire_context(3), gss_inquire_cred(3), gss_release_cred(3), gss_release_oid_set(3)

STANDARDS

RFC 2743 Generic Security Service Application Program Interface Version 2, Update 1

RFC 2744 Generic Security Service API Version 2 : C-bindings

HISTORY

The **gss_add_cred** function first appeared in FreeBSD 7.0.

AUTHORS

John Wray, Iris Associates

COPYRIGHT