NAME

krb5_mk_req, **krb5_mk_req_exact**, **krb5_mk_req_extended**, **krb5_rd_req**, **krb5_rd_req_with_keyblock**, **krb5_mk_rep**, **krb5_mk_rep_exact**, **krb5_mk_rep_extended**, **krb5_rd_rep**, **krb5_build_ap_req**, **krb5_verify_ap_req -** create and read application authentication request

LIBRARY

Kerberos 5 Library (libkrb5, -lkrb5)

SYNOPSIS

#include <krb5.h>

krb5_error_code
krb5_mk_req(*krb5_context context*, *krb5_auth_context *auth_context*, *const krb5_flags ap_req_options*, *const char *service, const char *hostname, krb5_data *in_data, krb5_ccache ccache*, *krb5_data *outbuf*);

krb5_error_code
krb5_mk_req_extended(*krb5_context context*, *krb5_auth_context *auth_context*, *const krb5_flags ap_req_options*, *krb5_data *in_data*, *krb5_creds *in_creds*, *krb5_data *outbuf*);

krb5_error_code
krb5_rd_req(*krb5_context context*, *krb5_auth_context *auth_context, const krb5_data *inbuf*, *krb5_const_principal server*, *krb5_keytab keytab*, *krb5_flags *ap_req_options*, *krb5_ticket **ticket*);

krb5_error_code
krb5_build_ap_req(*krb5_context context*, *krb5_enctype enctype*, *krb5_creds *cred*, *krb5_flags ap_options*, *krb5_data authenticator*, *krb5_data *retdata*);

krb5_error_code
krb5_verify_ap_req(*krb5_context context*, *krb5_auth_context *auth_context*, *krb5_ap_req *ap_req*, *krb5_const_principal server*, *krb5_keyblock *keyblock*, *krb5_flags flags*, *krb5_flags *ap_req_options*, *krb5_ticket **ticket*);

DESCRIPTION

The functions documented in this manual page document the functions that facilitates the exchange between a Kerberos client and server. They are the core functions used in the authentication exchange between the client and the server.

The **krb5_mk_req** and **krb5_mk_req_extended** creates the Kerberos message KRB_AP_REQ that is sent from the client to the server as the first packet in a client/server exchange. The result that should be sent to server is stored in *outbuf*.

auth_context should be allocated with **krb5_auth_con_init**() or NULL passed in, in that case, it will be allocated and freed internally.

The input data *in_data* will have a checksum calculated over it and checksum will be transported in the message to the server.

ap_req_options can be set to one or more of the following flags:

AP_OPTS_USE_SESSION_KEY
    Use the session key when creating the request, used for user to user authentication.

AP_OPTS_MUTUAL_REQUIRED
    Mark the request as mutual authenticate required so that the receiver returns a mutual authentication packet.

The **krb5_rd_req** read the AP_REQ in *inbuf* and verify and extract the content. If *server* is specified, that server will be fetched from the *keytab* and used unconditionally. If *server* is NULL, the *keytab* will be search for a matching principal.

The *keytab* argument specifies what keytab to search for receiving principals. The arguments *ap_req_options* and *ticket* returns the content.

When the AS-REQ is a user to user request, neither of *keytab* or *principal* are used, instead **krb5_rd_req**() expects the session key to be set in *auth_context*.

The **krb5_verify_ap_req** and **krb5_build_ap_req** both constructs and verify the AP_REQ message, should not be used by external code.

SEE ALSO

krb5(3), krb5.conf(5)