

NAME

`krb5_get_init_creds`, `krb5_get_init_creds_keytab`, `krb5_get_init_creds_opt`,
`krb5_get_init_creds_opt_alloc`, `krb5_get_init_creds_opt_free`, `krb5_get_init_creds_opt_init`,
`krb5_get_init_creds_opt_set_address_list`, `krb5_get_init_creds_opt_set_addressless`,
`krb5_get_init_creds_opt_set_anonymous`, `krb5_get_init_creds_opt_set_default_flags`,
`krb5_get_init_creds_opt_set_etype_list`, `krb5_get_init_creds_opt_set_forwardable`,
`krb5_get_init_creds_opt_set_pa_password`, `krb5_get_init_creds_opt_set_paq_request`,
`krb5_get_init_creds_opt_set_preauth_list`, `krb5_get_init_creds_opt_set_proxiable`,
`krb5_get_init_creds_opt_set_renew_life`, `krb5_get_init_creds_opt_set_salt`,
`krb5_get_init_creds_opt_set_tkt_life`, `krb5_get_init_creds_opt_set_canonicalize`,
`krb5_get_init_creds_opt_set_win2k`, `krb5_get_init_creds_password`, `krb5_prompt`,
`krb5_prompter_posix` - Kerberos 5 initial authentication functions

LIBRARY

Kerberos 5 Library (`libkrb5`, `-lkrb5`)

SYNOPSIS

```
#include <krb5.h>
```

```
krb5_get_init_creds_opt;
```

```
krb5_error_code
```

```
krb5_get_init_creds_opt_alloc(krb5_context context, krb5_get_init_creds_opt **opt);
```

```
void
```

```
krb5_get_init_creds_opt_free(krb5_context context, krb5_get_init_creds_opt *opt);
```

```
void
```

```
krb5_get_init_creds_opt_init(krb5_get_init_creds_opt *opt);
```

```
void
```

```
krb5_get_init_creds_opt_set_address_list(krb5_get_init_creds_opt *opt, krb5_addresses *addresses);
```

```
void
```

```
krb5_get_init_creds_opt_set_addressless(krb5_get_init_creds_opt *opt, krb5_boolean addressless);
```

```
void
```

```
krb5_get_init_creds_opt_set_anonymous(krb5_get_init_creds_opt *opt, int anonymous);
```

```
void
```

krb5_get_init_creds_opt_set_default_flags(*krb5_context context, const char *appname, krb5_const_realm realm, krb5_get_init_creds_opt *opt*);

void

krb5_get_init_creds_opt_set_etype_list(*krb5_get_init_creds_opt *opt, krb5_etype *etype_list, int etype_list_length*);

void

krb5_get_init_creds_opt_set_forwardable(*krb5_get_init_creds_opt *opt, int forwardable*);

krb5_error_code

krb5_get_init_creds_opt_set_pa_password(*krb5_context context, krb5_get_init_creds_opt *opt, const char *password, krb5_s2k_proc key_proc*);

krb5_error_code

krb5_get_init_creds_opt_set_paq_request(*krb5_context context, krb5_get_init_creds_opt *opt, krb5_boolean req_pac*);

krb5_error_code

krb5_get_init_creds_opt_set_pkinit(*krb5_context context, krb5_get_init_creds_opt *opt, const char *cert_file, const char *key_file, const char *x509_anchors, int flags, char *password*);

void

krb5_get_init_creds_opt_set_preauth_list(*krb5_get_init_creds_opt *opt, krb5_preauthtype *preauth_list, int preauth_list_length*);

void

krb5_get_init_creds_opt_set_proxiability(*krb5_get_init_creds_opt *opt, int proxiability*);

void

krb5_get_init_creds_opt_set_renew_life(*krb5_get_init_creds_opt *opt, krb5_deltat renew_life*);

void

krb5_get_init_creds_opt_set_salt(*krb5_get_init_creds_opt *opt, krb5_data *salt*);

void

krb5_get_init_creds_opt_set_tkt_life(*krb5_get_init_creds_opt *opt, krb5_deltat tkt_life*);

krb5_error_code

krb5_get_init_creds_opt_set_canonicalize(*krb5_context context, krb5_get_init_creds_opt *opt,*

krb5_boolean req);

krb5_error_code

krb5_get_init_creds_opt_set_win2k(*krb5_context context, krb5_get_init_creds_opt *opt, krb5_boolean req*);

krb5_error_code

krb5_get_init_creds(*krb5_context context, krb5_creds *creds, krb5_principal client, krb5_prompter_fct prompter, void *prompter_data, krb5_deltat start_time, const char *in_tkt_service, krb5_get_init_creds_opt *options*);

krb5_error_code

krb5_get_init_creds_password(*krb5_context context, krb5_creds *creds, krb5_principal client, const char *password, krb5_prompter_fct prompter, void *prompter_data, krb5_deltat start_time, const char *in_tkt_service, krb5_get_init_creds_opt *in_options*);

krb5_error_code

krb5_get_init_creds_keytab(*krb5_context context, krb5_creds *creds, krb5_principal client, krb5_keytab keytab, krb5_deltat start_time, const char *in_tkt_service, krb5_get_init_creds_opt *options*);

int

krb5_prompter_posix(*krb5_context context, void *data, const char *name, const char *banner, int num_prompts, krb5_prompt prompts[]*);

DESCRIPTION

Getting initial credential ticket for a principal. That may include changing an expired password, and doing preauthentication. This interface that replaces the deprecated *krb5_in_tkt* and *krb5_in_cred* functions.

If you only want to verify a username and password, consider using *krb5_verify_user*(3) instead, since it also verifies that initial credentials with using a keytab to make sure the response was from the KDC.

First a *krb5_get_init_creds_opt* structure is initialized with **krb5_get_init_creds_opt_alloc**() or **krb5_get_init_creds_opt_init**(). **krb5_get_init_creds_opt_alloc**() allocates a extendible structures that needs to be freed with **krb5_get_init_creds_opt_free**(). The structure may be modified by any of the **krb5_get_init_creds_opt_set**() functions to change request parameters and authentication information.

If the caller want to use the default options, NULL can be passed instead.

The the actual request to the KDC is done by any of the **krb5_get_init_creds()**, **krb5_get_init_creds_password()**, or **krb5_get_init_creds_keytab()** functions. **krb5_get_init_creds()** is the least specialized function and can, with the right in data, behave like the latter two. The latter two are there for compatibility with older releases and they are slightly easier to use.

krb5_prompt is a structure containing the following elements:

```
typedef struct {
    const char *prompt;
    int hidden;
    krb5_data *reply;
    krb5_prompt_type type
} krb5_prompt;
```

prompt is the prompt that should shown to the user. If *hidden* is set, the prompter function shouldn't echo the output to the display device. *reply* must be preallocated; it will not be allocated by the prompter function. Possible values for the *type* element are:

```
KRB5_PROMPT_TYPE_PASSWORD
KRB5_PROMPT_TYPE_NEW_PASSWORD
KRB5_PROMPT_TYPE_NEW_PASSWORD_AGAIN
KRB5_PROMPT_TYPE_PREAUTH
KRB5_PROMPT_TYPE_INFO
```

krb5_prompter_posix() is the default prompter function in a POSIX environment. It matches the *krb5_prompter_fct* and can be used in the *krb5_get_init_creds* functions. **krb5_prompter_posix()** doesn't require *prompter_data*.

If the *start_time* is zero, then the requested ticket will be valid beginning immediately. Otherwise, the *start_time* indicates how far in the future the ticket should be postdated.

If the *in_tkt_service* name is non-NULL, that principal name will be used as the server name for the initial ticket request. The realm of the name specified will be ignored and will be set to the realm of the client name. If no *in_tkt_service* name is specified, *krbtgt/CLIENT-REALM@CLIENT-REALM* will be used.

For the rest of arguments, a configuration or library default will be used if no value is specified in the options structure.

krb5_get_init_creds_opt_set_address_list() sets the list of *addresses* that is should be stored in the ticket.

krb5_get_init_creds_opt_set_addressless() controls if the ticket is requested with addresses or not, **krb5_get_init_creds_opt_set_address_list()** overrides this option.

krb5_get_init_creds_opt_set_anonymous() make the request anonymous if the *anonymous* parameter is non-zero.

krb5_get_init_creds_opt_set_default_flags() sets the default flags using the configuration file.

krb5_get_init_creds_opt_set_etype_list() set a list of enctypees that the client is willing to support in the request.

krb5_get_init_creds_opt_set_forwardable() request a forwardable ticket.

krb5_get_init_creds_opt_set_pa_password() set the *password* and *key_proc* that is going to be used to get a new ticket. *password* or *key_proc* can be NULL if the caller wants to use the default values. If the *password* is unset and needed, the user will be prompted for it.

krb5_get_init_creds_opt_set_paq_request() sets the password that is going to be used to get a new ticket.

krb5_get_init_creds_opt_set_preauth_list() sets the list of client-supported preauth types.

krb5_get_init_creds_opt_set_proxiable() makes the request proxiable.

krb5_get_init_creds_opt_set_renew_life() sets the requested renewable lifetime.

krb5_get_init_creds_opt_set_salt() sets the salt that is going to be used in the request.

krb5_get_init_creds_opt_set_tkt_life() sets requested ticket lifetime.

krb5_get_init_creds_opt_set_canonicalize() requests that the KDC canonicalize the client principal if possible.

krb5_get_init_creds_opt_set_win2k() turns on compatibility with Windows 2000.

SEE ALSO

krb5(3), krb5_creds(3), krb5_verify_user(3), krb5.conf(5), kerberos(8)