

NAME

krb5_get_credentials, **krb5_get_credentials_with_flags**, **krb5_get_kdc_cred**, **krb5_get_renewed_creds** - get credentials from the KDC using `krbtgt`

LIBRARY

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

SYNOPSIS

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

```
krb5_error_code
```

```
krb5_get_credentials(krb5_context context, krb5_flags options, krb5_ccache ccache,
krb5_creds *in_creds, krb5_creds **out_creds);
```

```
krb5_error_code
```

```
krb5_get_credentials_with_flags(krb5_context context, krb5_flags options, krb5_kdc_flags flags,
krb5_ccache ccache, krb5_creds *in_creds, krb5_creds **out_creds);
```

```
krb5_error_code
```

```
krb5_get_kdc_cred(krb5_context context, krb5_ccache id, krb5_kdc_flags flags,
krb5_addresses *addresses, Ticket *second_ticket, krb5_creds *in_creds, krb5_creds **out_creds);
```

```
krb5_error_code
```

```
krb5_get_renewed_creds(krb5_context context, krb5_creds *creds, krb5_const_principal client,
krb5_ccache ccache, const char *in_tkt_service);
```

DESCRIPTION

krb5_get_credentials_with_flags() get credentials specified by `in_creds->server` and `in_creds->client` (the rest of the `in_creds` structure is ignored) by first looking in the `ccache` and if doesn't exist or is expired, fetch the credential from the KDC using the `krbtgt` in `ccache`. The credential is returned in `out_creds` and should be freed using the function **krb5_free_creds**().

Valid flags to pass into `options` argument are:

- `KRB5_GC_CACHED` Only check the `ccache`, don't go out on network to fetch credential.
- `KRB5_GC_USER_USER` Request a user to user ticket. This option doesn't store the resulting user to user credential in the `ccache`.
- `KRB5_GC_EXPIRED_OK` returns the credential even if it is expired, default behavior is trying to refetch the credential from the KDC.

Flags are KDCOptions, note the caller must fill in the bit-field and not use the integer associated structure.

krb5_get_credentials() works the same way as **krb5_get_credentials_with_flags()** except that the *flags* field is missing.

krb5_get_kdc_cred() does the same as the functions above, but the caller must fill in all the information and it's closer to the wire protocol.

krb5_get_renewed_creds() renews a credential given by *in_tkt_service* (if NULL the default krbtgt) using the credential cache *ccache*. The result is stored in *creds* and should be freed using *krb5_free_creds*.

EXAMPLES

Here is an example function that gets a credential from a credential cache *id* or the KDC and returns it to the caller.

```
#include <krb5.h>

int
getcred(krb5_context context, krb5_ccache id, krb5_creds **creds)
{
    krb5_error_code ret;
    krb5_creds in;

    ret = krb5_parse_name(context, "client@EXAMPLE.COM",
                          &in.client);

    if (ret)
        krb5_err(context, 1, ret, "krb5_parse_name");

    ret = krb5_parse_name(context, "host/server.example.com@EXAMPLE.COM",
                          &in.server);

    if (ret)
        krb5_err(context, 1, ret, "krb5_parse_name");

    ret = krb5_get_credentials(context, 0, id, &in, creds);
    if (ret)
        krb5_err(context, 1, ret, "krb5_get_credentials");

    return 0;
}
```

}

SEE ALSO

krb5(3), krb5_get_forwarded_creds(3), krb5.conf(5)