## NAME

pam_krb5 − Kerberos 5 authentication

## DESCRIPTION

pam_krb5.so reads its configuration information from the **appdefaults** section of **krb5.conf**(5). You should read the **krb5.conf**(5) man page before continuing here. The module expects its configuration information to be in the **pam** subsection of the **appdefaults** section.

## DIRECTIVES

Directives which take a *true*, *false*, or a PAM service name can also be selectively disabled for specific PAM services using the related "no_" option (exceptions to "debug = true" can be made using "no_debug", for example).

debug = *true|false|service [...]*

turns on debugging via **syslog**(3). Debug messages are logged with priority *LOG_DEBUG*.

debug_sensitive = *true|false|service [...]*

turns on debugging of sensitive information via **syslog**(3). Debug messages are logged with priority *LOG_DEBUG*.

addressless = *true|false|service [...]*

if set, requests a TGT with no address information. This can be necessary if you are using Kerberos through a NAT, or on systems whose IP addresses change regularly. This directive is deprecated in favor of the **libdefaults noaddresses** directive.

afs_cells = *cell.example.com [...]*

tells pam_krb5.so to obtain tokens for the listed cells, in addition to the local cell and the cell which contains the user's home directory, for the user. The module will guess the principal name of the AFS service for the listed cells, or it can be specified by listing cells in the form *cellname*=**principalname**.

banner = *Kerberos 5*

specifies what sort of password the module claims to be changing whenever it is called upon to change passwords. The default is **Kerberos 5**.

ccache_dir = */var/tmp*

specifies the directory in which to place credential cache files. The default is */tmp*.

ccname_template = *KEYRING:krb5cc_%U_%P*

ccname_template = *FILE:%d/krb5cc_%U_XXXXXX*

specifies the location in which to place the user's session-specific credential cache. This value is treated as a template, and these sequences are substituted:

| | |
|---|---|
| %u | login name |
| %U | login UID |
| %p | principal name |
| %r | realm name |
| %h | home directory |
| %d | the default ccache directory (as set with *ccache_dir*) |
| %P | the current process ID |
| %% | literal '%' |

The default is *FILE:%d/krb5cc_%U_XXXXXX*".

chpw_prompt = *true|false|service [...]*
tells pam_krb5.so to allow expired passwords to be changed during authentication attempts. While this is the traditional behavior exhibited by "kinit", it is inconsistent with the behavior expected by PAM, which expects authentication to (appear to) succeed, only to have password expiration be flagged by a subsequent call to the account management function. Some applications which don't handle password expiration correctly will fail unconditionally if the user's password is expired, and this flag can be used to attempt to work around this bug in those applications. The default is **false**.

cred_session=*true|false|service [...]*
specifies that pam_krb5 should create and destroy credential caches, as it does when the calling application opens and closes a PAM session, when the calling application establishes and deletes PAM credentials. This is done to compensate for applications which expect to create a credential cache but which don't use PAM session management. It is usually a harmless redundancy in applications which don't require it, so this option is enabled by default except for this list of services: "*sshd*".

existing_ticket = *true|false|service [...]*
tells pam_krb5.so to accept the presence of pre-existing Kerberos credentials provided by the calling application in the default credential cache as sufficient to authenticate the user, and to skip any account management checks. The default is **false**.

DANGER! Unless validation is also in use, it is relatively easy to produce a credential cache which looks "good enough" to fool pam_krb5.so.

external = *true|false|sshd ftp [...]*
tells pam_krb5.so to use Kerberos credentials provided by the calling application during session setup. This is most often useful for obtaining AFS tokens. The default is "*sshd sshd-rekey*".

forwardable = *true|false|service [...]*
controls whether or not credentials are forwardable. This directive is deprecated in favor of the **libdefaults forwardable** directive.

hosts = *hostname [...]*
specifies which other hosts credentials obtained by pam_krb5 will be good on. If your host is behind a firewall, you should add the IP address or name that the *KDC* sees it as to this list. This directive is deprecated in favor of the **libdefaults extra_addresses** directive.

ignore_afs=*true|false|service [...]*
tells pam_krb5.so to completely ignore the presence of AFS, preventing any attempts to obtain new tokens on behalf of the calling application.

ignore_unknown_principals=*true|false|service [...]*

ignore_unknown_spn=*true|false|service [...]*

ignore_unknown_upn=*true|false|service [...]*
specifies which other not pam_krb5 should return a PAM_IGNORE code to libpam instead of PAM_USER_UNKNOWN for users for whom the determined principal name is expired or does not exist.

initial_prompt=*true*|*false*|*service [...]*
tells pam_krb5.so whether or not to ask for a password before attempting authentication. If one is needed and pam_krb5.so has not prompted for it, the Kerberos library should trigger a request for a password.

keytab = *FILE:/etc/krb5.keytab*

keytab = *FILE:/etc/krb5.keytab imap=FILE:/etc/imap.keytab*
specifies the name of a keytab file to search for a service key for use in validating TGTs. The location can be specified on a per-service basis by specifying a list of locations in the form *pam_service=location*. The default is *FILE:/etc/krb5.keytab*.

mappings = *regex1 regex2 [...]*
specifies that pam_krb5 should derive the user's principal name from the Unix user name by first checking if the user name matches **regex1**, and formulating a principal name using **regex2**. For example, "**mappings = EXAMPLE\(.*) $1@EXAMPLE.COM**" would map any user with a name of the form "EXAMPLE\whatever" to a principal name of "whatever@EXAMPLE.COM". This is primarily targeted at allowing pam_krb5 to be used to authenticate users whose user information is provided by **winbindd**(8). This will frequently require the reverse to be configured by setting up an auth_to_local rule elsewhere in **krb5.conf**(5).

minimum_uid = *0*
specifies the minimum UID of users being authenticated. If a user with a UID less than this value attempts authentication, the request will be ignored.

multiple_ccaches=*true*|*false*|*service [...]*
specifies that pam_krb5 should maintain multiple credential caches for applications that both set credentials and open a PAM session, but which set the KRB5CCNAME variable after doing only one of the two. This option is usually not necessary for most services.

preauth_options =
controls the preauthentication options which pam_krb5 passes to libkrb5, if the system-defaults need to be overridden. The list is treated as a template, and these sequences are substituted:

| | |
|---|---|
| %u | login name |
| %U | login UID |
| %p | principal name |
| %r | realm name |
| %h | home directory |
| %d | the default ccache directory (as set with *ccache_dir*) |
| %P | the current process ID |
| %% | literal '%' |

proxiable = *true*|*false*|*service [...]*
controls whether or not credentials are proxiable. If not specified, they are. This directive is deprecated in favor of the **libdefaults proxiable** directive.

null_afs=*true*|*false*|*service [...]*
tells pam_krb5.so, when it attempts to set tokens, to try to get credentials for services with names which resemble afs@*REALM* before attempting to get credentials for services with names resembling afs/*cell@REALM*. The default is to assume that the cell's name is the instance in the AFS service's Kerberos principal name.

pwhelp = *filename*
specifies the name of a text file whose contents will be displayed to clients who attempt to change their passwords. There is no default.

renew_lifetime = *36000*
default renewable lifetime, in seconds. This specifies how much time you have after getting credentials to renew them. This directive is deprecated in favor of the **libdefaults renew_lifetime** directive.

subsequent_prompt = *true|false|service [...]*
controls whether or not pam_krb5.so will allow the Kerberos library to ask the user for a password or other information, if the previously-entered password is somehow insufficient for authenticating the user. This is commonly needed to allow a user to log in when that user's password has expired. The default is **true**.

If the calling application does not properly support PAM conversations (possibly due to limitations of a network protocol which it is serving), this may be need to be disabled for that application to prevent it from supplying the user's current password in a password-changing situations when a new password is called for.

ticket_lifetime = *36000*
default credential lifetime, in seconds.

tokens = *true|false|service [...]*
signals that pam_krb5.so should create an AFS PAG and obtain tokens during authentication in addition to session setup. This is primarily useful in server applications which need to access a user's files but which do not open PAM sessions before doing so. For correctly-written applications, this flag is not necessary.

token_strategy = *rxk5,2b[,...]*
controls how, and using which format, pam_krb5.so should attept to set AFS tokens for the user's session. By default, the module is configured with "token_strategy = *v4,524,2b,rxk5*". Recognized strategy names include:
*rxk5* rxk5 (requires OpenAFS 1.6 or later)
*2b* rxkad "2b" (requires OpenAFS 1.2.8 or later)

use_shmem = *true|false|service [...]*
tells pam_krb5.so to pass credentials from the authentication service function to the session management service function using shared memory for specific services. By default, the module is configured with "use_shmem = *sshd*".

validate = *true|false|service [...]*
specifies whether or not to attempt validation of the TGT using the local keytab. The default is **true**. The **libdefaults verify_ap_req_nofail** setting can affect whether or not errors reading the keytab which are encountered during validation will be suppressed.

## EXAMPLE

```
[appdefaults]
  pam = {
    ticket_lifetime = 36000
    renew_lifetime = 36000
```

```
  forwardable = true
  validate = true
  ccache_dir = /var/tmp
  external = sshd
  tokens = imap ftpd
  TEST.EXAMPLE.COM = {
   debug = true
   afs_cells = testcell.example.com othercell.example.com
   keytab = FILE:/etc/krb5.keytab httpd=FILE:/etc/httpd.keytab
  }
 }
```

## FILES

*/etc/krb5.conf*

## SEE ALSO

**pam_krb5**(8)

## BUGS

Probably, but let's hope not. If you find any, please file them in the bug database at http://bugzilla.redhat.com/ against the "pam_krb5" component.

## AUTHOR

Nalin Dahyabhai <nalin@redhat.com>