NAME

pam_krb5 - Kerberos 5 PAM module

SYNOPSIS

/usr/lib/pam_krb5.so

DESCRIPTION

The Kerberos 5 service module for PAM, typically */usr/lib/pam_krb5.so*, provides functionality for three PAM categories: authentication, account management, and password management. It also provides null functions for session management. The *pam_krb5.so* module is a shared object that can be dynamically loaded to provide the necessary functionality upon demand. Its path is specified in the PAM configuration file.

Kerberos 5 Authentication Module

The Kerberos 5 authentication component provides functions to verify the identity of a user (**pam_sm_authenticate**()) and to set user specific credentials (**pam_sm_setcred**()). **pam_sm_authenticate**() converts the supplied username into a Kerberos principal, by appending the default local realm name. It also supports usernames with explicit realm names. If a realm name is supplied, then upon a successful return, it changes the username by mapping the principal name into a local username (calling **krb5_aname_to_localname**()). This typically just means the realm name is stripped.

It prompts the user for a password and obtains a new Kerberos TGT for the principal. The TGT is verified by obtaining a service ticket for the local host.

When prompting for the current password, the authentication module will use the prompt "Password for <principal>:".

The **pam_sm_setcred**() function stores the newly acquired credentials in a credentials cache, and sets the environment variable KRB5CCNAME appropriately. The credentials cache should be destroyed by the user at logout with kdestroy(1).

The following options may be passed to the authentication module:

debug syslog(3) debugging information at LOG_DEBUG level.

no_warn suppress warning messages to the user. These messages include reasons why the user’s authentication attempt was declined.

use_first_pass If the authentication module is not the first in the stack, and a previous module obtained the user's password, that password is used to authenticate the user. If this fails, the authentication module returns failure without prompting the user for a password. This option has no effect if the authentication module is the first in the stack, or if no previous modules obtained the user's password.

try_first_pass This option is similar to the **use_first_pass** option, except that if the previously obtained password fails, the user is prompted for another password.

forwardable Obtain forwardable Kerberos credentials for the user.

no_ccache Do not save the obtained credentials in a credentials cache. This is a useful option if the authentication module is used for services such as ftp or pop, where the user would not be able to destroy them. [This is not a recommendation to use the module for those services.]

ccache=*name* Use *name* as the credentials cache. *name* must be in the form *type*:*residual*. The special tokens '%u', to designate the decimal UID of the user; and '%p', to designate the current process ID; can be used in *name*.

allow_kdc_spoof

Allow **pam_krb5** to succeed even if there is no host or service key available in a keytab to authenticate the Kerberos KDC's ticket. If there is no such key, for example on a host with no keytabs, **pam_krb5** will fail immediately without prompting the user.

Warning: If the host has not been configured with a keytab from the KDC, setting this option makes it vulnerable to malicious KDCs, e.g. via DNS flooding, because **pam_krb5** has no way to distinguish the legitimate KDC from a spoofed KDC.

no_user_check

Do not verify if a user exists on the local system. This option implies the **no_ccache** option because there is no secure local uid/gid for the cache file.

Kerberos 5 Account Management Module

The Kerberos 5 account management component provides a function to perform account management, **pam_sm_acct_mgmt**(). The function verifies that the authenticated principal is allowed to login to the local user account by calling **krb5_kuserok**() (which checks the user's *.k5login* file).

Kerberos 5 Password Management Module

The Kerberos 5 password management component provides a function to change passwords (**pam_sm_chauthtok**()). The username supplied (the user running the passwd(1) command, or the username given as an argument) is mapped into a Kerberos principal name, using the same technique as in the authentication module. Note that if a realm name was explicitly supplied during authentication, but not during a password change, the mapping done by the password management module may not result in the same principal as was used for authentication.

Unlike when changing a UNIX password, the password management module will allow any user to change any principal's password (if the user knows the principal's old password, of course). Also unlike UNIX, root is always prompted for the principal's old password.

The password management module uses the same heuristics as kpasswd(1) to determine how to contact the Kerberos password server.

The following options may be passed to the password management module:

debug syslog(3) debugging information at LOG_DEBUG level.

use_first_pass If the password management module is not the first in the stack, and a previous module obtained the user's old password, that password is used to authenticate the user. If this fails, the password management module returns failure without prompting the user for the old password. If successful, the new password entered to the previous module is also used as the new Kerberos password. If the new password fails, the password management module returns failure without prompting the user for a new password.

try_first_pass This option is similar to the **use_first_pass** option, except that if the previously obtained old or new passwords fail, the user is prompted for them.

Kerberos 5 Session Management Module

The Kerberos 5 session management component provides functions to initiate (**pam_sm_open_session**()) and terminate (**pam_sm_close_session**()) sessions. Since session management is not defined under Kerberos 5, both of these functions simply return success. They are provided only because of the naming conventions for PAM modules.

ENVIRONMENT

KRB5CCNAME Location of the credentials cache.

FILES

/tmp/krb5cc_uid	default credentials cache (*uid* is the decimal UID of the user).
$HOME/.k5login	file containing Kerberos principals that are allowed access.

SEE ALSO

kdestroy(1), passwd(1), syslog(3), pam.conf(5), pam(3)

NOTES

Applications should not call **pam_authenticate**() more than once between calls to **pam_start**() and **pam_end**() when using the Kerberos 5 PAM module.