NAME

pam_get_authtok - retrieve authentication token

SYNOPSIS

#include <sys/types.h>
#include <security/pam_appl.h>

int
pam_get_authtok(*pam_handle_t *pamh*, *int item*, *const char **authtok*, *const char *prompt*);

DESCRIPTION

The **pam_get_authtok**() function either prompts the user for an authentication token or retrieves a cached authentication token, depending on circumstances. Either way, a pointer to the authentication token is stored in the location pointed to by the *authtok* argument, and the corresponding PAM item is updated.

The *item* argument must have one of the following values:

PAM_AUTHTOK Returns the current authentication token, or the new token when changing authentication tokens.

PAM_OLDAUTHTOK
Returns the previous authentication token when changing authentication tokens.

The *prompt* argument specifies a prompt to use if no token is cached. If it is NULL, the PAM_AUTHTOK_PROMPT or PAM_OLDAUTHTOK_PROMPT item, as appropriate, will be used. If that item is also NULL, a hardcoded default prompt will be used. Additionally, when **pam_get_authtok**() is called from a service module, the prompt may be affected by module options as described below. The prompt is then expanded using openpam_subst(3) before it is passed to the conversation function.

If *item* is set to PAM_AUTHTOK and there is a non-null PAM_OLDAUTHTOK item, **pam_get_authtok**() will ask the user to confirm the new token by retyping it. If there is a mismatch, **pam_get_authtok**() will return PAM_TRY_AGAIN.

MODULE OPTIONS

When called by a service module, **pam_get_authtok**() will recognize the following module options:

authtok_prompt Prompt to use when *item* is set to PAM_AUTHTOK. This option overrides both the *prompt* argument and the PAM_AUTHTOK_PROMPT item.

echo_pass
: If the application's conversation function allows it, this lets the user see what they are typing. This should only be used for non-reusable authentication tokens.

oldauthtok_prompt
: Prompt to use when *item* is set to PAM_OLDAUTHTOK. This option overrides both the *prompt* argument and the PAM_OLDAUTHTOK_PROMPT item.

try_first_pass
: If the requested item is non-null, return it without prompting the user. Typically, the service module will verify the token, and if it does not match, clear the item before calling **pam_get_authtok**() a second time.

use_first_pass
: Do not prompt the user at all; just return the cached value, or PAM_AUTH_ERR if there is none.

RETURN VALUES

The **pam_get_authtok**() function returns one of the following values:

[PAM_SUCCESS] Success.

[PAM_BAD_CONSTANT]
Bad constant.

[PAM_BAD_ITEM] Unrecognized or restricted item.

[PAM_BUF_ERR] Memory buffer error.

[PAM_CONV_ERR] Conversation failure.

[PAM_SYSTEM_ERR]
System error.

[PAM_TRY_AGAIN]
Try again.

SEE ALSO

openpam_get_option(3), openpam_subst(3), pam(3), pam_conv(3), pam_get_item(3), pam_get_user(3), pam_strerror(3)

STANDARDS

The **pam_get_authtok**() function is an OpenPAM extension.

AUTHORS

The **pam_get_authtok**() function and this manual page were developed for the FreeBSD Project by ThinkSec AS and Network Associates Laboratories, the Security Research Division of Network Associates, Inc. under DARPA/SPAWAR contract N66001-01-C-8035 ("CBOSS"), as part of the DARPA CHATS research program.

The OpenPAM library is maintained by Dag-Erling Smørgrav <*des@des.no*>.