

NAME

pam_conv - PAM conversation system

LIBRARY

Pluggable Authentication Module Library (libpam, -lpam)

SYNOPSIS

```
#include <security/pam_appl.h>
```

```
struct pam_message {
    int    msg_style;
    char  *msg;
};

struct pam_response {
    char  *resp;
    int   resp_retcode;
};

struct pam_conv {
    int   (*conv)(int, const struct pam_message **,
                struct pam_response **, void *);
    void  *appdata_ptr;
};
```

DESCRIPTION

The PAM library uses an application-defined callback to communicate with the user. This callback is specified by the *struct pam_conv* passed to **pam_start()** at the start of the transaction. It is also possible to set or change the conversation function at any point during a PAM transaction by changing the value of the PAM_CONV item.

The conversation function's first argument specifies the number of messages (up to PAM_MAX_NUM_MSG) to process. The second argument is a pointer to an array of pointers to *pam_message* structures containing the actual messages.

Each message can have one of four types, specified by the *msg_style* member of *struct pam_message*:

PAM_PROMPT_ECHO_OFF

Display a prompt and accept the user's response without echoing it to the terminal. This is commonly used for passwords.

PAM_PROMPT_ECHO_ON

Display a prompt and accept the user's response, echoing it to the terminal. This is commonly used for login names and one-time passphrases.

PAM_ERROR_MSG Display an error message.

PAM_TEXT_INFO Display an informational message.

In each case, the prompt or message to display is pointed to by the *msg* member of *struct pam_message*. It can be up to **PAM_MAX_MSG_SIZE** characters long, including the terminating NUL.

On success, the conversation function should allocate and fill a contiguous array of *struct pam_response*, one for each message that was passed in. A pointer to the user's response to each message (or NULL in the case of informational or error messages) should be stored in the *resp* member of the corresponding *struct pam_response*. Each response can be up to **PAM_MAX_RESP_SIZE** characters long, including the terminating NUL.

The *resp_retcode* member of *struct pam_response* is unused and should be set to zero.

The conversation function should store a pointer to this array in the location pointed to by its third argument. It is the caller's responsibility to release both this array and the responses themselves, using `free(3)`. It is the conversation function's responsibility to ensure that it is legal to do so.

The *appdata_ptr* member of *struct pam_conv* is passed unmodified to the conversation function as its fourth and final argument.

On failure, the conversation function should release any resources it has allocated, and return one of the predefined PAM error codes.

RETURN VALUES

The conversation function should return one of the following values:

[PAM_BUF_ERR] Memory buffer error.

[PAM_CONV_ERR] Conversation failure.

[PAM_SUCCESS] Success.

[PAM_SYSTEM_ERR]
System error.

SEE ALSO

openpam_nullconv(3), openpam_ttyconv(3), pam(3), pam_error(3), pam_get_item(3), pam_info(3), pam_prompt(3), pam_set_item(3), pam_start(3)

STANDARDS

X/Open Single Sign-On Service (XSSO) - Pluggable Authentication Modules, June 1997.

AUTHORS

The OpenPAM library 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 Smórgrav <des@des.no>.