NAME

pam_listfile − deny or allow services based on an arbitrary file

SYNOPSIS

pam_listfile.so item=[tty|user|rhost|ruser|group|shell] sense=[allow|deny] file=*/path/filename* onerr=[succeed|fail] [apply=[*user*|@*group*]] [quiet]

DESCRIPTION

pam_listfile is a PAM module which provides a way to deny or allow services based on an arbitrary file.

The module gets the **item** of the type specified −− *user* specifies the username, *PAM_USER*; tty specifies the name of the terminal over which the request has been made, *PAM_TTY*; rhost specifies the name of the remote host (if any) from which the request was made, *PAM_RHOST*; and ruser specifies the name of the remote user (if available) who made the request, *PAM_RUSER* −− and looks for an instance of that item in the **file=***filename*. filename contains one line per item listed. If the item is found, then if **sense=***allow*, *PAM_SUCCESS* is returned, causing the authorization request to succeed; else if **sense=***deny*, *PAM_AUTH_ERR* is returned, causing the authorization request to fail.

If an error is encountered (for instance, if filename does not exist, or a poorly−constructed argument is encountered), then if *onerr=succeed*, *PAM_SUCCESS* is returned, otherwise if *onerr=fail*, *PAM_AUTH_ERR* or *PAM_SERVICE_ERR* (as appropriate) will be returned.

An additional argument, **apply=**, can be used to restrict the application of the above to a specific user (**apply=***username*) or a given group (**apply=@***groupname*). This added restriction is only meaningful when used with the *tty*, *rhost* and *shell* items.

Besides this last one, all arguments should be specified; do not count on any default behavior.

No credentials are awarded by this module.

OPTIONS

item=[tty|user|rhost|ruser|group|shell]
What is listed in the file and should be checked for.

sense=[allow|deny]
Action to take if found in file, if the item is NOT found in the file, then the opposite action is requested.

file=*/path/filename*
File containing one item per line. The file needs to be a plain file and not world writable.

onerr=[succeed|fail]
What to do if something weird happens like being unable to open the file.

apply=[*user***|@***group***]**
Restrict the user class for which the restriction apply. Note that with **item=[user|ruser|group]** this does not make sense, but for **item=[tty|rhost|shell]** it have a meaning.

quiet
Do not treat service refusals or missing list files as errors that need to be logged.

MODULE TYPES PROVIDED

All module types (**auth**, **account**, **password** and **session**) are provided.

RETURN VALUES

PAM_AUTH_ERR
Authentication failure.

PAM_BUF_ERR
Memory buffer error.

PAM_IGNORE
The rule does not apply to the **apply** option.

PAM_SERVICE_ERR

Error in service module.

PAM_SUCCESS
Success.

EXAMPLES

Classic 'ftpusers' authentication can be implemented with this entry in /etc/pam.d/ftpd:

```
#
# deny ftp−access to users listed in the /etc/ftpusers file
#
auth    required      pam_listfile.so \
      onerr=succeed item=user sense=deny file=/etc/ftpusers
```

Note, users listed in /etc/ftpusers file are (counterintuitively) *not* allowed access to the ftp service.

To allow login access only for certain users, you can use a /etc/pam.d/login entry like this:

```
#
# permit login to users listed in /etc/loginusers
#
auth    required      pam_listfile.so \
      onerr=fail item=user sense=allow file=/etc/loginusers
```

For this example to work, all users who are allowed to use the login service should be listed in the file /etc/loginusers. Unless you are explicitly trying to lock out root, make sure that when you do this, you leave a way for root to log in, either by listing root in /etc/loginusers, or by listing a user who is able to *su* to the root account.

SEE ALSO

pam.conf(5), **pam.d**(5), **pam**(7)

AUTHOR

pam_listfile was written by Michael K. Johnson <johnsonm@redhat.com> and Elliot Lee <sopwith@cuc.edu>.