NAME

pam_faillock − Module counting authentication failures during a specified interval

SYNOPSIS

auth ... pam_faillock.so {preauth|authfail|authsucc} [conf=*/path/to/config−file*]
[dir=*/path/to/tally−directory*] [even_deny_root] [deny=*n*] [fail_interval=*n*]
[unlock_time=*n*] [root_unlock_time=*n*] [admin_group=*name*] [audit] [silent]
[no_log_info]

account ... pam_faillock.so [dir=*/path/to/tally−directory*] [no_log_info]

DESCRIPTION

This module maintains a list of failed authentication attempts per user during a specified interval and locks the account in case there were more than *deny* consecutive failed authentications.

Normally, failed attempts to authenticate *root* will **not** cause the root account to become blocked, to prevent denial−of−service: if your users aren't given shell accounts and root may only login via **su** or at the machine console (not telnet/rsh, etc), this is safe.

OPTIONS

{preauth|authfail|authsucc}

This argument must be set accordingly to the position of this module instance in the PAM stack.

The *preauth* argument must be used when the module is called before the modules which ask for the user credentials such as the password. The module just examines whether the user should be blocked from accessing the service in case there were anomalous number of failed consecutive authentication attempts recently. This call is optional if *authsucc* is used.

The *authfail* argument must be used when the module is called after the modules which determine the authentication outcome, failed. Unless the user is already blocked due to previous authentication failures, the module will record the failure into the appropriate user tally file.

The *authsucc* argument must be used when the module is called after the modules which determine the authentication outcome, succeeded. Unless the user is already blocked due to previous authentication failures, the module will then clear the record of the failures in the respective user tally file. Otherwise it will return authentication error. If this call is not done, the pam_faillock will not distinguish between consecutive and non−consecutive failed authentication attempts. The *preauth* call must be used in such case. Due to complications in the way the PAM stack can be configured it is also possible to call *pam_faillock* as an account module. In such configuration the module must be also called in the *preauth* stage.

conf=/path/to/config−file

Use another configuration file instead of the default /etc/security/faillock.conf.

The options for configuring the module behavior are described in the **faillock.conf**(5) manual page. The options specified on the module command line override the values from the configuration file.

MODULE TYPES PROVIDED

The **auth** and **account** module types are provided.

RETURN VALUES

PAM_AUTH_ERR

An invalid option was given, the module was not able to retrieve the user name, no valid counter file was found, or too many failed logins.

PAM_BUF_ERR

Memory buffer error.

PAM_CONV_ERR

The conversation method supplied by the application failed to obtain the username.

PAM_INCOMPLETE

The conversation method supplied by the application returned PAM_CONV_AGAIN.

PAM_SUCCESS
Everything was successful.

PAM_IGNORE
User not present in passwd database.

NOTES

Configuring options on the module command line is not recommend. The /etc/security/faillock.conf should be used instead.

The setup of *pam_faillock* in the PAM stack is different from the *pam_tally2* module setup.

Individual files with the failure records are created as owned by the user. This allows **pam_faillock.so** module to work correctly when it is called from a screensaver.

Note that using the module in **preauth** without the **silent** option specified in /etc/security/faillock.conf or with *requisite* control field leaks an information about existence or non−existence of an user account in the system because the failures are not recorded for the unknown users. The message about the user account being locked is never displayed for non−existing user accounts allowing the adversary to infer that a particular account is not existing on a system.

EXAMPLES

Here are two possible configuration examples for /etc/pam.d/login. They make *pam_faillock* to lock the account after 4 consecutive failed logins during the default interval of 15 minutes. Root account will be locked as well. The accounts will be automatically unlocked after 20 minutes.

In the first example the module is called only in the *auth* phase and the module does not print any information about the account being blocked by *pam_faillock*. The *preauth* call can be added to tell users that their logins are blocked by the module and also to abort the authentication without even asking for password in such case.

/etc/security/faillock.conf file example:

```
deny=4
unlock_time=1200
silent
```

/etc/pam.d/config file example:

```
auth     required       pam_securetty.so
auth     required       pam_env.so
auth     required       pam_nologin.so
# optionally call: auth requisite pam_faillock.so preauth
# to display the message about account being locked
auth     [success=1 default=bad] pam_unix.so
auth     [default=die]  pam_faillock.so authfail
auth     sufficient     pam_faillock.so authsucc
auth     required       pam_deny.so
account  required       pam_unix.so
password required       pam_unix.so shadow
session  required       pam_selinux.so close
session  required       pam_loginuid.so
session  required       pam_unix.so
session  required       pam_selinux.so open
```

In the second example the module is called both in the *auth* and *account* phases and the module informs the authenticating user when the account is locked if **silent** option is not specified in the faillock.conf.

```
auth     required       pam_securetty.so
auth     required       pam_env.so
auth     required       pam_nologin.so
auth     required       pam_faillock.so preauth
# optionally use requisite above if you do not want to prompt for the password
# on locked accounts
auth     sufficient     pam_unix.so
auth     [default=die]  pam_faillock.so authfail
auth     required       pam_deny.so
account  required       pam_faillock.so
# if you drop the above call to pam_faillock.so the lock will be done also
# on non−consecutive authentication failures
account  required       pam_unix.so
password required       pam_unix.so shadow
session  required       pam_selinux.so close
session  required       pam_loginuid.so
session  required       pam_unix.so
session  required       pam_selinux.so open
```

FILES

/var/run/faillock/*

the files logging the authentication failures for users

/etc/security/faillock.conf

the config file for pam_faillock options

SEE ALSO

faillock(8), **faillock.conf**(5), **pam.conf**(5), **pam.d**(5), **pam**(8)

AUTHOR

pam_faillock was written by Tomas Mraz.