NAME

pam_tally2 − The login counter (tallying) module

SYNOPSIS

pam_tally2.so [file=*/path/to/counter*] [onerr=[*fail*|*succeed*]] [magic_root] [even_deny_root] [deny=*n*] [lock_time=*n*] [unlock_time=*n*] [root_unlock_time=*n*] [serialize] [audit] [silent] [no_log_info] [debug]

pam_tally2 [−−file */path/to/counter*] [−−user *username*] [−−reset[=*n*]] [−−quiet]

DESCRIPTION

This module maintains a count of attempted accesses, can reset count on success, can deny access if too many attempts fail.

pam_tally2 comes in two parts: **pam_tally2.so** and **pam_tally2**. The former is the PAM module and the latter, a stand−alone program. **pam_tally2** is an (optional) application which can be used to interrogate and manipulate the counter file. It can display user counts, set individual counts, or clear all counts. Setting artificially high counts may be useful for blocking users without changing their passwords. For example, one might find it useful to clear all counts every midnight from a cron job.

Normally, failed attempts to access *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

GLOBAL OPTIONS

This can be used for *auth* and *account* module types.

onerr=[*fail***|***succeed***]**
If something weird happens (like unable to open the file), return with **PAM_SUCCESS** if **onerr=***succeed* is given, else with the corresponding PAM error code.

file=*/path/to/counter*
File where to keep counts. Default is /var/log/tallylog.

audit
Will log the user name into the system log if the user is not found.

silent
Don't print informative messages.

no_log_info
Don't log informative messages via **syslog**(3).

debug
Always log tally count when it is incremented as a debug level message to the system log.

AUTH OPTIONS

Authentication phase first increments attempted login counter and checks if user should be denied access. If the user is authenticated and the login process continues on call to **pam_setcred**(3) it resets the attempts counter.

deny=*n*
Deny access if tally for this user exceeds *n*.

lock_time=*n*
Always deny for *n* seconds after failed attempt.

unlock_time=*n*
Allow access after *n* seconds after failed attempt. If this option is used the user will be locked out for the specified amount of time after he exceeded his maximum allowed attempts. Otherwise the account is locked until the lock is removed by a manual intervention of the system administrator.

magic_root
If the module is invoked by a user with uid=0 the counter is not incremented. The sysadmin

should use this for user launched services, like **su**, otherwise this argument should be omitted.

even_deny_root
Root account can become unavailable.

root_unlock_time=*n*
This option implies **even_deny_root** option. Allow access after *n* seconds to root account after failed attempt. If this option is used the root user will be locked out for the specified amount of time after he exceeded his maximum allowed attempts.

serialize
Serialize access to the tally file using locks. This option might be used only for non−multithreaded services because it depends on the fcntl locking of the tally file. Also it is a good idea to use this option only in such configurations where the time between auth phase and account or setcred phase is not dependent on the authenticating client. Otherwise the authenticating client will be able to prevent simultaneous authentications by the same user by simply artificially prolonging the time the file record lock is held.

ACCOUNT OPTIONS
Account phase resets attempts counter if the user is **not** magic root. This phase can be used optionally for services which don't call **pam_setcred**(3) correctly or if the reset should be done regardless of the failure of the account phase of other modules.

magic_root
If the module is invoked by a user with uid=0 the counter is not changed. The sysadmin should use this for user launched services, like **su**, otherwise this argument should be omitted.

MODULE TYPES PROVIDED

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

RETURN VALUES

PAM_AUTH_ERR
A 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_SUCCESS
Everything was successful.

PAM_USER_UNKNOWN
User not known.

NOTES

pam_tally2 is not compatible with the old pam_tally faillog file format. This is caused by requirement of compatibility of the tallylog file format between 32bit and 64bit architectures on multiarch systems.

There is no setuid wrapper for access to the data file such as when the **pam_tally2.so** module is called from xscreensaver. As this would make it impossible to share PAM configuration with such services the following workaround is used: If the data file cannot be opened because of insufficient permissions (**EACCES**) the module returns **PAM_IGNORE**.

EXAMPLES

Add the following line to /etc/pam.d/login to lock the account after 4 failed logins. Root account will be locked as well. The accounts will be automatically unlocked after 20 minutes. The module does not have to be called in the account phase because the **login** calls **pam_setcred**(3) correctly.

```
auth     required      pam_securetty.so
auth     required      pam_tally2.so deny=4 even_deny_root unlock_time=1200
auth     required      pam_env.so
auth     required      pam_unix.so
auth     required      pam_nologin.so
account  required      pam_unix.so
password required      pam_unix.so
```

```
session  required     pam_limits.so
session  required     pam_unix.so
session  required     pam_lastlog.so nowtmp
session  optional     pam_mail.so standard
```

FILES

/var/log/tallylog
failure count logging file

SEE ALSO

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

AUTHOR

pam_tally2 was written by Tim Baverstock and Tomas Mraz.