NAME

login.defs − shadow password suite configuration

DESCRIPTION

The /etc/login.defs file defines the site−specific configuration for the shadow password suite. This file is required. Absence of this file will not prevent system operation, but will probably result in undesirable operation.

This file is a readable text file, each line of the file describing one configuration parameter. The lines consist of a configuration name and value, separated by whitespace. Blank lines and comment lines are ignored. Comments are introduced with a "#" pound sign and the pound sign must be the first non−white character of the line.

Parameter values may be of four types: strings, booleans, numbers, and long numbers. A string is comprised of any printable characters. A boolean should be either the value *yes* or *no*. An undefined boolean parameter or one with a value other than these will be given a *no* value. Numbers (both regular and long) may be either decimal values, octal values (precede the value with *0*) or hexadecimal values (precede the value with *0x*). The maximum value of the regular and long numeric parameters is machine−dependent.

Please note that the parameters in this configuration file control the behavior of the tools from the shadow−utils component. None of these tools uses the PAM mechanism, and the utilities that use PAM (such as the passwd command) should be configured elsewhere. The only values that affect PAM modules are **FAIL_DELAY** for pam_faildelay module and **UMASK** for pam_umask module. Refer to pam(8) for more information.

The following configuration items are provided:

CHFN_AUTH (boolean)
If *yes*, the **chfn** program will require authentication before making any changes, unless run by the superuser.

CHFN_RESTRICT (string)
This parameter specifies which values in the *gecos* field of the /etc/passwd file may be changed by regular users using the **chfn** program. It can be any combination of letters *f*, *r*, *w*, *h*, for Full name, Room number, Work phone, and Home phone, respectively. For backward compatibility, *yes* is equivalent to *rwh* and *no* is equivalent to *frwh*. If not specified, only the superuser can make any changes. The most restrictive setting is better achieved by not installing **chfn** SUID.

CHSH_AUTH (boolean)
If *yes*, the **chsh** program will require authentication before making any changes, unless run by the superuser.

CONSOLE (string)
If defined, either full pathname of a file containing device names (one per line) or a ":" delimited list of device names. Root logins will be allowed only upon these devices.

If not defined, root will be allowed on any device.

The device should be specified without the /dev/ prefix.

CONSOLE_GROUPS (string)
List of groups to add to the user´s supplementary groups set when logging in on the console (as determined by the CONSOLE setting). Default is none.

Use with caution − it is possible for users to gain permanent access to these groups, even when not logged in on the console.

CREATE_HOME (boolean)
Indicate if a home directory should be created by default for new users.

This setting does not apply to system users, and can be overriden on the command line.

DEFAULT_HOME (boolean)
Indicate if login is allowed if we can´t cd to the home directory. Default in no.

If set to *yes*, the user will login in the root (/) directory if it is not possible to cd to her home directory.

ENCRYPT_METHOD (string)
This defines the system default encryption algorithm for encrypting passwords (if no algorithm are specified on the command line).

It can take one of these values:

- *DES* (default)
- *MD5*
- *SHA256*
- *SHA512*

Note: this parameter overrides the **MD5_CRYPT_ENAB** variable.

ENV_HZ (string)
If set, it will be used to define the HZ environment variable when a user login. The value must be preceded by *HZ=*. A common value on Linux is *HZ=100*.

ENV_PATH (string)
If set, it will be used to define the PATH environment variable when a regular user login. The value can be preceded by *PATH=*, or a colon separated list of paths (for example */bin:/usr/bin*). The default value is *PATH=/bin:/usr/bin*.

ENV_SUPATH (string)
If set, it will be used to define the PATH environment variable when the superuser login. The value can be preceded by *PATH=*, or a colon separated list of paths (for example */sbin:/bin:/usr/sbin:/usr/bin*). The default value is *PATH=/bin:/usr/bin*.

ENV_TZ (string)
If set, it will be used to define the TZ environment variable when a user login. The value can be the name of a timezone preceded by *TZ=* (for example *TZ=CST6CDT*), or the full path to the file containing the timezone specification (for example /etc/tzname).

If a full path is specified but the file does not exist or cannot be read, the default is to use *TZ=CST6CDT*.

ENVIRON_FILE (string)
If this file exists and is readable, login environment will be read from it. Every line should be in the form name=value.

Lines starting with a # are treated as comment lines and ignored.

ERASECHAR (number)
Terminal ERASE character (*010* = backspace, *0177* = DEL).

The value can be prefixed "0" for an octal value, or "0x" for an hexadecimal value.

FAIL_DELAY (number)
Delay in seconds before being allowed another attempt after a login failure.

FAILLOG_ENAB (boolean)
Enable logging and display of /var/log/faillog login failure info.

FAKE_SHELL (string)
If set, **login** will execute this shell instead of the users´ shell specified in /etc/passwd.

FTMP_FILE (string)
If defined, login failures will be logged in this file in a utmp format.

GID_MAX (number), **GID_MIN** (number)
Range of group IDs used for the creation of regular groups by **useradd**, **groupadd**, or **newusers**.

HUSHLOGIN_FILE (string)
If defined, this file can inhibit all the usual chatter during the login sequence. If a full pathname is specified, then hushed mode will be enabled if the user´s name or shell are found in the file. If not a full pathname, then hushed mode will be enabled if the file exists in the user´s home directory.

ISSUE_FILE (string)
If defined, this file will be displayed before each login prompt.

KILLCHAR (number)
Terminal KILL character (*025* = CTRL/U).

The value can be prefixed "0" for an octal value, or "0x" for an hexadecimal value.

LASTLOG_ENAB (boolean)
Enable logging and display of /var/log/lastlog login time info.

LOG_OK_LOGINS (boolean)
Enable logging of successful logins.

LOG_UNKFAIL_ENAB (boolean)
Enable display of unknown usernames when login failures are recorded.

Note: logging unknown usernames may be a security issue if an user enter her password instead of her login name.

LOGIN_RETRIES (number)
Maximum number of login retries in case of bad password.

LOGIN_STRING (string)
The string used for prompting a password. The default is to use "Password: ", or a translation of that string. If you set this variable, the prompt will no be translated.

If the string contains *%s*, this will be replaced by the user´s name.

LOGIN_TIMEOUT (number)
Max time in seconds for login.

MAIL_CHECK_ENAB (boolean)
Enable checking and display of mailbox status upon login.

You should disable it if the shell startup files already check for mail ("mailx −e" or equivalent).

MAIL_DIR (string)
The mail spool directory. This is needed to manipulate the mailbox when its corresponding user account is modified or deleted. If not specified, a compile−time default is used.

MAIL_FILE (string)
Defines the location of the users mail spool files relatively to their home directory.

The **MAIL_DIR** and **MAIL_FILE** variables are used by **useradd**, **usermod**, and **userdel** to create, move, or delete the user´s mail spool.

If **MAIL_CHECK_ENAB** is set to *yes*, they are also used to define the **MAIL** environment variable.

MAX_MEMBERS_PER_GROUP (number)
Maximum members per group entry. When the maximum is reached, a new group entry (line) is started in /etc/group (with the same name, same password, and same GID).

The default value is 0, meaning that there are no limits in the number of members in a group.

This feature (split group) permits to limit the length of lines in the group file. This is useful to make sure that lines for NIS groups are not larger than 1024 characters.

If you need to enforce such limit, you can use 25.

Note: split groups may not be supported by all tools (even in the Shadow toolsuite). You should not use this variable unless you really need it.

MD5_CRYPT_ENAB (boolean)
Indicate if passwords must be encrypted using the MD5−based algorithm. If set to *yes*, new passwords will be encrypted using the MD5−based algorithm compatible with the one used by recent releases of FreeBSD. It supports passwords of unlimited length and longer salt strings. Set to *no* if you need to copy encrypted passwords to other systems which don´t understand the new algorithm. Default is *no*.

This variable is superceded by the **ENCRYPT_METHOD** variable or by any command line option used to configure the encryption algorithm.

This variable is deprecated. You should use **ENCRYPT_METHOD**.

MOTD_FILE (string)
If defined, ":" delimited list of "message of the day" files to be displayed upon login.

NOLOGINS_FILE (string)
If defined, name of file whose presence will inhibit non−root logins. The contents of this file should be a message indicating why logins are inhibited.

OBSCURE_CHECKS_ENAB (boolean)
Enable additional checks upon password changes.

PASS_ALWAYS_WARN (boolean)
Warn about weak passwords (but still allow them) if you are root.

PASS_CHANGE_TRIES (number)
Maximum number of attempts to change password if rejected (too easy).

PASS_MAX_DAYS (number)
The maximum number of days a password may be used. If the password is older than this, a password change will be forced. If not specified, −1 will be assumed (which disables the restriction).

PASS_MIN_DAYS (number)
The minimum number of days allowed between password changes. Any password changes attempted sooner than this will be rejected. If not specified, −1 will be assumed (which disables the restriction).

PASS_WARN_AGE (number)
The number of days warning given before a password expires. A zero means warning is given only upon the day of expiration, a negative value means no warning is given. If not specified, no warning will be provided.

PASS_MAX_DAYS, **PASS_MIN_DAYS** and **PASS_WARN_AGE** are only used at the time of account creation. Any changes to these settings won´t affect existing accounts.

PASS_MAX_LEN (number), **PASS_MIN_LEN** (number)
Number of significant characters in the password for crypt(). **PASS_MAX_LEN** is 8 by default. Don´t change unless your crypt() is better. This is ignored if **MD5_CRYPT_ENAB** set to *yes*.

PORTTIME_CHECKS_ENAB (boolean)

Enable checking of time restrictions specified in /etc/porttime.

QUOTAS_ENAB (boolean)
Enable setting of ulimit, umask, and niceness from passwd gecos field.

SHA_CRYPT_MIN_ROUNDS (number), **SHA_CRYPT_MAX_ROUNDS** (number)
When **ENCRYPT_METHOD** is set to *SHA256* or *SHA512*, this defines the number of SHA rounds used by the encryption algorithm by default (when the number of rounds is not specified on the command line).

With a lot of rounds, it is more difficult to brute forcing the password. But note also that more CPU resources will be needed to authenticate users.

If not specified, the libc will choose the default number of rounds (5000).

The values must be inside the 1000−999999999 range.

If only one of the **SHA_CRYPT_MIN_ROUNDS** or **SHA_CRYPT_MAX_ROUNDS** values is set, then this value will be used.

If **SHA_CRYPT_MIN_ROUNDS** > **SHA_CRYPT_MAX_ROUNDS**, the highest value will be used.

SULOG_FILE (string)
If defined, all su activity is logged to this file.

SU_NAME (string)
If defined, the command name to display when running "su −". For example, if this is defined as "su" then a "ps" will display the command is "−su". If not defined, then "ps" would display the name of the shell actually being run, e.g. something like "−sh".

SU_WHEEL_ONLY (boolean)
If *yes*, the user must be listed as a member of the first gid 0 group in /etc/group (called *root* on most Linux systems) to be able to **su** to uid 0 accounts. If the group doesn´t exist or is empty, no one will be able to **su** to uid 0.

SYS_GID_MAX (number), **SYS_GID_MIN** (number)
Range of group IDs used for the creation of system groups by **useradd**, **groupadd**, or **newusers**.

SYS_UID_MAX (number), **SYS_UID_MIN** (number)
Range of user IDs used for the creation of system users by **useradd** or **newusers**.

SYSLOG_SG_ENAB (boolean)
Enable "syslog" logging of **sg** activity.

SYSLOG_SU_ENAB (boolean)
Enable "syslog" logging of **su** activity − in addition to sulog file logging.

TTYGROUP (string), **TTYPERM** (string)
The terminal permissions: the login tty will be owned by the **TTYGROUP** group, and the permissions will be set to **TTYPERM**.

By default, the ownership of the terminal is set to the user´s primary group and the permissions are set to *0600*.

TTYGROUP can be either the name of a group or a numeric group identifier.

If you have a **write** program which is "setgid" to a special group which owns the terminals, define TTYGROUP to the group number and TTYPERM to 0620. Otherwise leave TTYGROUP

commented out and assign TTYPERM to either 622 or 600.

TTYTYPE_FILE (string)
If defined, file which maps tty line to TERM environment parameter. Each line of the file is in a format something like "vt100 tty01".

UID_MAX (number), **UID_MIN** (number)
Range of user IDs used for the creation of regular users by **useradd** or **newusers**.

ULIMIT (number)
Default **ulimit** value.

UMASK (number)
The file mode creation mask is initialized to this value. If not specified, the mask will be initialized to 022.

useradd and **newusers** use this mask to set the mode of the home directory they create

It is also used by **login** to define users´ initial umask. Note that this mask can be overriden by the user´s GECOS line (if **QUOTAS_ENAB** is set) or by the specification of a limit with the *K* identifier in **limits**(5).

USERDEL_CMD (string)
If defined, this command is run when removing a user. It should remove any at/cron/print jobs etc. owned by the user to be removed (passed as the first argument).

The return code of the script is not taken into account.

Here is an example script, which removes the user´s cron, at and print jobs:

```
#! /bin/sh

# Check for the required argument.
if [ $# != 1 ]; then
        echo "Usage: $0 username"
        exit 1
fi

# Remove cron jobs.
crontab −r −u $1

# Remove at jobs.
# Note that it will remove any jobs owned by the same UID,
# even if it was shared by a different username.
AT_SPOOL_DIR=/var/spool/cron/atjobs
find $AT_SPOOL_DIR −name "[^.]*" −type f −user $1 −delete \;

# Remove print jobs.
lprm $1

# All done.
exit 0
```

USERGROUPS_ENAB (boolean)
Enable setting of the umask group bits to be the same as owner bits (examples: 022 −> 002, 077

−> 007) for non−root users, if the uid is the same as gid, and username is the same as the primary group name.

If set to *yes*, **userdel** will remove the user´s group if it contains no more members, and **useradd** will create by default a group with the name of the user.

CROSS REFERENCES

The following cross references show which programs in the shadow password suite use which parameters.

chfn

CHFN_AUTH CHFN_RESTRICT LOGIN_STRING

chgpasswd
ENCRYPT_METHOD MAX_MEMBERS_PER_GROUP MD5_CRYPT_ENAB SHA_CRYPT_MAX_ROUNDS SHA_CRYPT_MIN_ROUNDS

chpasswd
ENCRYPT_METHOD MD5_CRYPT_ENAB SHA_CRYPT_MAX_ROUNDS SHA_CRYPT_MIN_ROUNDS

chsh
CHSH_AUTH LOGIN_STRING

gpasswd
ENCRYPT_METHOD MAX_MEMBERS_PER_GROUP MD5_CRYPT_ENAB SHA_CRYPT_MAX_ROUNDS SHA_CRYPT_MIN_ROUNDS

groupadd
GID_MAX GID_MIN MAX_MEMBERS_PER_GROUP SYS_GID_MAX SYS_GID_MIN

groupdel
MAX_MEMBERS_PER_GROUP

groupmems
MAX_MEMBERS_PER_GROUP

groupmod
MAX_MEMBERS_PER_GROUP

grpck
MAX_MEMBERS_PER_GROUP

grpconv
MAX_MEMBERS_PER_GROUP

grpunconv
MAX_MEMBERS_PER_GROUP

login

CONSOLE CONSOLE_GROUPS DEFAULT_HOME ENV_HZ ENV_PATH ENV_SUPATH ENV_TZ ENVIRON_FILE ERASECHAR FAIL_DELAY FAILLOG_ENAB FAKE_SHELL FTMP_FILE HUSHLOGIN_FILE ISSUE_FILE KILLCHAR LASTLOG_ENAB LOGIN_RETRIES LOGIN_STRING LOGIN_TIMEOUT LOG_OK_LOGINS LOG_UNKFAIL_ENAB MAIL_CHECK_ENAB MAIL_DIR MAIL_FILE MOTD_FILE NOLOGINS_FILE PORTTIME_CHECKS_ENAB QUOTAS_ENAB TTYGROUP TTYPERM TTYTYPE_FILE ULIMIT UMASK USERGROUPS_ENAB

newgrp / sg
SYSLOG_SG_ENAB

newusers
ENCRYPT_METHOD GID_MAX GID_MIN MAX_MEMBERS_PER_GROUP

MD5_CRYPT_ENAB PASS_MAX_DAYS PASS_MIN_DAYS PASS_WARN_AGE SHA_CRYPT_MAX_ROUNDS SHA_CRYPT_MIN_ROUNDS SYS_GID_MAX SYS_GID_MIN SYS_UID_MAX SYS_UID_MIN UID_MAX UID_MIN UMASK

passwd
ENCRYPT_METHOD MD5_CRYPT_ENAB OBSCURE_CHECKS_ENAB PASS_ALWAYS_WARN PASS_CHANGE_TRIES PASS_MAX_LEN PASS_MIN_LEN SHA_CRYPT_MAX_ROUNDS SHA_CRYPT_MIN_ROUNDS

pwck
PASS_MAX_DAYS PASS_MIN_DAYS PASS_WARN_AGE

pwconv
PASS_MAX_DAYS PASS_MIN_DAYS PASS_WARN_AGE

su

CONSOLE CONSOLE_GROUPS DEFAULT_HOME ENV_HZ ENVIRON_FILE ENV_PATH ENV_SUPATH ENV_TZ LOGIN_STRING MAIL_CHECK_ENAB MAIL_DIR MAIL_FILE QUOTAS_ENAB SULOG_FILE SU_NAME SU_WHEEL_ONLY SYSLOG_SU_ENAB USERGROUPS_ENAB

sulogin
ENV_HZ ENV_TZ

useradd
CREATE_HOME GID_MAX GID_MIN MAIL_DIR MAX_MEMBERS_PER_GROUP PASS_MAX_DAYS PASS_MIN_DAYS PASS_WARN_AGE SYS_GID_MAX SYS_GID_MIN SYS_UID_MAX SYS_UID_MIN UID_MAX UID_MIN UMASK

userdel
MAIL_DIR MAIL_FILE MAX_MEMBERS_PER_GROUP USERDEL_CMD USERGROUPS_ENAB

usermod
MAIL_DIR MAIL_FILE MAX_MEMBERS_PER_GROUP

SEE ALSO

login(1), **passwd**(1), **su**(1), **passwd**(5), **shadow**(5), **pam**(8).