

**NAME**

limits.conf – configuration file for the pam\_limits module

**DESCRIPTION**

The *pam\_limits.so* module applies ulimit limits, nice priority and number of simultaneous login sessions limit to user login sessions. This description of the configuration file syntax applies to the */etc/security/limits.conf* file and *\*.conf* files in the */etc/security/limits.d* directory.

The syntax of the lines is as follows:

```
<domain><type><item><value>
```

The fields listed above should be filled as follows:

**<domain>**

- a username
- a groupname, with **@group** syntax. This should not be confused with netgroups.
- the wildcard *\**, for default entry.
- the wildcard *%*, for maxlogins limit only, can also be used with **%group** syntax. If the *%* wildcard is used alone it is identical to using *\** with maxsyslogins limit. With a group specified after *%* it limits the total number of logins of all users that are member of the group.
- an uid range specified as *<min\_uid>:<max\_uid>*. If *min\_uid* is omitted, the match is exact for the *max\_uid*. If *max\_uid* is omitted, all uids greater than or equal *min\_uid* match.
- a gid range specified as *@<min\_gid>:<max\_gid>*. If *min\_gid* is omitted, the match is exact for the *max\_gid*. If *max\_gid* is omitted, all gids greater than or equal *min\_gid* match. For the exact match all groups including the user's supplementary groups are examined. For the range matches only the user's primary group is examined.
- a gid specified as *%:<gid>* applicable to maxlogins limit only. It limits the total number of logins of all users that are member of the group with the specified gid.

**NOTE:** group and wildcard limits are not applied to the root user. To set a limit for the root user, this field must contain the literal username **root**.

**<type>****hard**

for enforcing **hard** resource limits. These limits are set by the superuser and enforced by the Kernel. The user cannot raise his requirement of system resources above such values.

**soft**

for enforcing **soft** resource limits. These limits are ones that the user can move up or down within the permitted range by any pre-existing **hard** limits. The values specified with this token can be thought of as *default* values, for normal system usage.

–

for enforcing both **soft** and **hard** resource limits together.

Note, if you specify a type of '-' but neglect to supply the item and value fields then the module will never enforce any limits on the specified user/group etc. .

**<item>****core**

limits the core file size (KB)

**data**

maximum data size (KB)

**fsize**

maximum filesize (KB)

**memlock**

maximum locked-in-memory address space (KB)

**nofile**

maximum number of open file descriptors

**rss**

maximum resident set size (KB) (Ignored in Linux 2.4.30 and higher)

**stack**

maximum stack size (KB)

**cpu**

maximum CPU time (minutes)

**nproc**

maximum number of processes

**as**

address space limit (KB)

**maxlogins**

maximum number of logins for this user (this limit does not apply to user with *uid=0*)

**maxsyslogins**

maximum number of all logins on system; user is not allowed to log-in if total number of all user logins is greater than specified number (this limit does not apply to user with *uid=0*)

**priority**

the priority to run user process with (negative values boost process priority)

**locks**

maximum locked files (Linux 2.4 and higher)

**sigpending**

maximum number of pending signals (Linux 2.6 and higher)

**msgqueue**

maximum memory used by POSIX message queues (bytes) (Linux 2.6 and higher)

**nice**

maximum nice priority allowed to raise to (Linux 2.6.12 and higher) values: [-20,19]

**rtprio**

maximum realtime priority allowed for non-privileged processes (Linux 2.6.12 and higher)

**chroot**

the directory to chroot the user to

All items support the values *-1*, *unlimited* or *infinity* indicating no limit, except for **priority** and **nice**.

If a hard limit or soft limit of a resource is set to a valid value, but outside of the supported range of the local system, the system may reject the new limit or unexpected behavior may occur. If the control value *required* is used, the module will reject the login if a limit could not be set.

In general, individual limits have priority over group limits, so if you impose no limits for *admin* group, but one of the members in this group have a limits line, the user will have its limits set according to this line.

Also, please note that all limit settings are set *per login*. They are not global, nor are they permanent; existing only for the duration of the session. One exception is the *maxlogin* option, this one is system wide. But there is a race, concurrent logins at the same time will not always be detect as such but only counted as one.

In the *limits* configuration file, the '#' character introduces a comment – after which the rest of the line is ignored.

The pam\_limits module does report configuration problems found in its configuration file and errors via **syslog(3)**.

### EXAMPLES

These are some example lines which might be specified in /etc/security/limits.conf.

```
*          soft  core      0
root       hard  core      100000
*          hard  nofile   512
@student   hard  nproc    20
@faculty   soft  nproc    20
@faculty   hard  nproc    50
ftp        hard  nproc    0
@student   -    maxlogins 4
:123       hard  cpu      5000
@500:      soft  cpu      10000
600:700    hard  locks    10
```

### SEE ALSO

**pam\_limits(8)**, **pam.d(5)**, **pam(7)**, **getrlimit(2)**, **getrlimit(3p)**

### AUTHOR

pam\_limits was initially written by Cristian Gafton <gafton@redhat.com>