NAME

namespace.conf − the namespace configuration file

DESCRIPTION

The *pam_namespace.so* module allows setup of private namespaces with polyinstantiated directories. Directories can be polyinstantiated based on user name or, in the case of SELinux, user name, sensitivity level or complete security context. If an executable script /etc/security/namespace.init exists, it is used to initialize the namespace every time an instance directory is set up and mounted. The script receives the polyinstantiated directory path and the instance directory path as its arguments.

The /etc/security/namespace.conf file specifies which directories are polyinstantiated, how they are polyinstantiated, how instance directories would be named, and any users for whom polyinstantiation would not be performed.

When someone logs in, the file namespace.conf is scanned. Comments are marked by # characters. Each non comment line represents one polyinstantiated directory. The fields are separated by spaces but can be quoted by *"* characters also escape sequences *b*, *n*, and *t* are recognized. The fields are as follows:

polydirinstance_prefixmethodlist_of_uids

The first field, *polydir*, is the absolute pathname of the directory to polyinstantiate. The special string *$HOME* is replaced with the user's home directory, and *$USER* with the username. This field cannot be blank.

The second field, *instance_prefix* is the string prefix used to build the pathname for the instantiation of <polydir>. Depending on the polyinstantiation *method* it is then appended with "instance differentiation string" to generate the final instance directory path. This directory is created if it did not exist already, and is then bind mounted on the <polydir> to provide an instance of <polydir> based on the <method> column. The special string *$HOME* is replaced with the user's home directory, and *$USER* with the username. This field cannot be blank.

The third field, *method*, is the method used for polyinstantiation. It can take these values; "user" for polyinstantiation based on user name, "level" for polyinstantiation based on process MLS level and user name, "context" for polyinstantiation based on process security context and user name, "tmpfs" for mounting tmpfs filesystem as an instance dir, and "tmpdir" for creating temporary directory as an instance dir which is removed when the user's session is closed. Methods "context" and "level" are only available with SELinux. This field cannot be blank.

The fourth field, *list_of_uids*, is a comma separated list of user names for whom the polyinstantiation is not performed. If left blank, polyinstantiation will be performed for all users. If the list is preceded with a single "~" character, polyinstantiation is performed only for users in the list.

The *method* field can contain also following optional flags separated by *:* characters.

create=mode,*owner*,*group* − create the polyinstantiated directory. The mode, owner and group parameters are optional. The default for mode is determined by umask, the default owner is the user whose session is opened, the default group is the primary group of the user.

iscript=path − path to the instance directory init script. The base directory for relative paths is /etc/security/namespace.d.

noinit − instance directory init script will not be executed.

shared − the instance directories for "context" and "level" methods will not contain the user name and will be shared among all users.

mntopts=value − value of this flag is passed to the mount call when the tmpfs mount is done. It allows for example the specification of the maximum size of the tmpfs instance that is created by the mount call. See **mount**(8) for details.

The directory where polyinstantiated instances are to be created, must exist and must have, by default, the mode of 0000. The requirement that the instance parent be of mode 0000 can be overridden with the command line option *ignore_instance_parent_mode*

In case of context or level polyinstantiation the SELinux context which is used for polyinstantiation is the context used for executing a new process as obtained by getexeccon. This context must be set by the calling application or pam_selinux.so module. If this context is not set the polyinstatiation will be based just on user name.

The "instance differentiation string" is <user name> for "user" method and <user name>_<raw directory context> for "context" and "level" methods. If the whole string is too long the end of it is replaced with md5sum of itself. Also when command line option *gen_hash* is used the whole string is replaced with md5sum of itself.

EXAMPLES

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

```
# The following three lines will polyinstantiate /tmp,
# /var/tmp and user's home directories. /tmp and /var/tmp
# will be polyinstantiated based on the security level
# as well as user name, whereas home directory will be
# polyinstantiated based on the full security context and user name.
# Polyinstantiation will not be performed for user root
# and adm for directories /tmp and /var/tmp, whereas home
# directories will be polyinstantiated for all users.
#
# Note that instance directories do not have to reside inside
# the polyinstantiated directory. In the examples below,
# instances of /tmp will be created in /tmp−inst directory,
# where as instances of /var/tmp and users home directories
# will reside within the directories that are being
# polyinstantiated.
#
/tmp     /tmp−inst/              level      root,adm
/var/tmp /var/tmp/tmp−inst/      level      root,adm
$HOME    $HOME/$USER.inst/inst− context
```

For the <service>s you need polyinstantiation (login for example) put the following line in /etc/pam.d/<service> as the last line for session group:

session required pam_namespace.so [arguments]

This module also depends on pam_selinux.so setting the context.

SEE ALSO

pam_namespace(8), **pam.d**(5), **pam**(7)

AUTHORS

The namespace.conf manual page was written by Janak Desai <janak@us.ibm.com>. More features added by Tomas Mraz <tmraz@redhat.com>.