

## NAME

`pam_namespace` – PAM module for configuring namespace for a session

## SYNOPSIS

```
pam_namespace.so [debug] [unmnt_remnt] [unmnt_only] [require_selinux] [gen_hash]
                  [ignore_config_error] [ignore_instance_parent_mode] [unmount_on_close]
                  [use_current_context] [use_default_context] [mount_private]
```

## DESCRIPTION

The `pam_namespace` PAM module sets up a private namespace for a session with polyinstantiated directories. A polyinstantiated directory provides a different instance of itself based on user name, or when using SELinux, user name, security context or both. If an executable script `/etc/security/namespace.init` exists, it is used to initialize the instance directory after it is set up and mounted on the polyinstantiated directory. The script receives the polyinstantiated directory path, the instance directory path, flag whether the instance directory was newly created (0 for no, 1 for yes), and the user name as its arguments.

The `pam_namespace` module disassociates the session namespace from the parent namespace. Any mounts/unmounts performed in the parent namespace, such as mounting of devices, are not reflected in the session namespace. To propagate selected mount/unmount events from the parent namespace into the disassociated session namespace, an administrator may use the special `shared-subtree` feature. For additional information on `shared-subtree` feature, please refer to the `mount(8)` man page and the `shared-subtree` description at <http://lwn.net/Articles/159077> and <http://lwn.net/Articles/159092>.

## OPTIONS

### **debug**

A lot of debug information is logged using syslog

### **unmnt\_remnt**

For programs such as `su` and `newrole`, the login session has already setup a polyinstantiated namespace. For these programs, polyinstantiation is performed based on new user id or security context, however the command first needs to undo the polyinstantiation performed by login. This argument instructs the command to first undo previous polyinstantiation before proceeding with new polyinstantiation based on new id/context

### **unmnt\_only**

For trusted programs that want to undo any existing bind mounts and process instance directories on their own, this argument allows them to unmount currently mounted instance directories

### **require\_selinux**

If `selinux` is not enabled, return failure

### **gen\_hash**

Instead of using the security context string for the instance name, generate and use its md5 hash.

### **ignore\_config\_error**

If a line in the configuration file corresponding to a polyinstantiated directory contains format error, skip that line process the next line. Without this option, `pam` will return an error to the calling program resulting in termination of the session.

### **ignore\_instance\_parent\_mode**

Instance parent directories by default are expected to have the restrictive mode of 000. Using this option, an administrator can choose to ignore the mode of the instance parent. This option should be used with caution as it will reduce security and isolation goals of the polyinstantiation mechanism.

### **unmount\_on\_close**

Explicitly unmount the polyinstantiated directories instead of relying on automatic namespace destruction after the last process in a namespace exits. This option should be used only in case it is ensured by other means that there cannot be any processes running in the private namespace left after the session close. It is also useful only in case there are multiple `pam` session calls in sequence from the same process.

### **use\_current\_context**

Useful for services which do not change the SELinux context with `setexeccon` call. The module will use the current SELinux context of the calling process for the level and context polyinstantiation.

#### **use\_default\_context**

Useful for services which do not use `pam_selinux` for changing the SELinux context with `setexeccon` call. The module will use the default SELinux context of the user for the level and context polyinstantiation.

#### **mount\_private**

This option can be used on systems where the `/` mount point or its submounts are made shared (for example with a `mount --make-rshared /` command). The module will mark the whole directory tree so any mount and unmount operations in the polyinstantiation namespace are private. Normally the `pam_namespace` will try to detect the shared `/` mount point and make the polyinstantiated directories private automatically. This option has to be used just when only a subtree is shared and `/` is not.

Note that mounts and unmounts done in the private namespace will not affect the parent namespace if this option is used or when the shared `/` mount point is autodetected.

### **MODULE TYPES PROVIDED**

Only the **session** module type is provided. The module must not be called from multithreaded processes.

### **RETURN VALUES**

**PAM\_SUCCESS**

Namespace setup was successful.

**PAM\_SERVICE\_ERR**

Unexpected system error occurred while setting up namespace.

**PAM\_SESSION\_ERR**

Unexpected namespace configuration error occurred.

### **FILES**

`/etc/security/namespace.conf`

Main configuration file

`/etc/security/namespace.d`

Directory for additional configuration files

`/etc/security/namespace.init`

Init script for instance directories

### **EXAMPLES**

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]
```

To use polyinstantiation with graphical display manager `gdm`, insert the following line, before `exit 0`, in `/etc/gdm/PostSession/Default`:

```
/usr/sbin/gdm-safe-restart
```

This allows `gdm` to restart after each session and appropriately adjust namespaces of display manager and the X server. If polyinstantiation of `/tmp` is desired along with the graphical environment, then additional configuration changes are needed to address the interaction of X server and font server namespaces with their use of `/tmp` to create communication sockets. Please use the initialization script `/etc/security/namespace.init` to ensure that the X server and its clients can appropriately access the communication socket `X0`. Please refer to the sample instructions provided in the comment section of the instance initialization script `/etc/security/namespace.init`. In addition, perform the following changes to use graphical environment with polyinstantiation of `/tmp`:

1. Disable the use of font server by commenting out "FontPath" line in `/etc/X11/xorg.conf`. If you do want to use the font server

then you will have to augment the instance initialization script to appropriately provide `/tmp/.font-unix` from the polyinstantiated `/tmp`.

2. Ensure that the `gdm` service is setup to use `pam_namespace`, as described above, by modifying `/etc/pam.d/gdm`.
3. Ensure that the display manager is configured to restart X server with each new session. This default setup can be verified by making sure that `/usr/share/gdm/defaults.conf` contains `"AlwaysRestartServer=true"`, and it is not overridden by `/etc/gdm/custom.conf`.

## SEE ALSO

**namespace.conf(5)**, **pam.d(5)**, **mount(8)**, **pam(7)**.

## AUTHORS

The namespace setup scheme was designed by Stephen Smalley, Janak Desai and Chad Sellers. The `pam_namespace` PAM module was developed by Janak Desai <janak@us.ibm.com>, Chad Sellers <csellers@tresys.com> and Steve Grubb <sgrubb@redhat.com>. Additional improvements by Xavier Toth <txtoth@gmail.com> and Tomas Mraz <tmraz@redhat.com>.