NAME

getprofattr, getprofnam, setprofattr, endprofattr – get profile description and attributes

SYNOPSIS

cc [*flag* ...] *file* ... **–lsecdb –lsocket –lnsl –lintl** [*library* ...]

#include <prof.h>

profattr_t *getprofattr(void);

profattr_t *getprofnam(const char *_name_);

void free_profattr(profattr_t *_pd_);

void setprofattr(void);

void endprofattr(void);

DESCRIPTION

The **getprofattr( )** and **getprofnam( )** functions each return a **prof_attr** entry. Entries can come from any of the sources specified in the **nsswitch.conf**(4) file.

The **getprofattr( )** function enumerates **prof_attr** entries. The **getprofnam( )** function searches for a **prof_attr** entry with a given *name*. Successive calls to these functions return either successive **prof_attr** entries or **NULL**.

The internal representation of a **prof_attr** entry is a **profattr_t** structure defined in **<prof_attr.h>** with the following members:

```
char  *name; /* Name of the profile */
char  *res1; /* Reserved for future use */
char  *res2; /* Reserved for future use */
char  *desc; /* Description/Purpose of the profile */
kva_t*attr; /* Profile attributes */
```

The **free_profattr( )** function call releases memory allocated by the **getprofattr** and **getprofnam** functions.

The **setprofattr( )** function "rewinds" to the beginning of the enumeration of **prof_attr** entries. Calls to **getprofnam( )** can leave the enumeration in an indeterminate state. Therefore, **setprofattr( )** should be called before the first call to **getprofattr( )**.

The **endprofattr( )** function may be called to indicate that **prof_attr** processing is complete; the system may then close any open **prof_attr** file, deallocate storage, and so forth.

RETURN VALUES

The **getprofattr( )** function returns a pointer to a **profattr_t** if it successfully enumerates an entry; otherwise it returns **NULL**, indicating the end of the enumeration.

The **getprofnam( )** function returns a pointer to a **profattr_t** if it successfully locates the requested entry; otherwise it returns **NULL**.

Individual attributes in the **prof_attr_t** structure can be referred to by calling the **kva_match**(3) function.

Because the list of legal keys is likely to expand, any code must be written to ignore unknown key-value pairs without error.

FILES **/etc/security/prof_attr** Stores profiles and their descriptions.

ATTRIBUTES See **attributes**(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	MT-Safe.

SEE ALSO **auths**(1)**, profiles**(1) **getexecattr**(3), **getauthattr**(3)**,** and **prof_attr**(4)**,**

NOTES **getprofattr( )** and **getprofnam( )** both allocate memory for the pointers they return. This memory should be deallocated with the **free_profattr()** call.

Programs that use the interfaces described in this manual page cannot be linked statically, since the implementations of these functions employ dynamic loading and linking of shared objects at run time. Note that these interfaces are reentrant even though they do not use the _r suffix naming convention.