NAME

getauthattr, getauthnam, setauthattr, endauthattr, chkauthattr – get authorization entry

SYNOPSIS

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

#include <auth_attr.h>

#include <secdb.h>

authattr_t *getauthattr(void);

authattr_t *getauthnam(const char ****name*);**

void free_authattr(authattr_t ****auth*);**

void setauthattr(void);

void endauthattr(void);

**int chkauthattr(const char ****authname,* **const char ****username);*

DESCRIPTION

The **getauthattr( )** and **getauthnam( )** functions each return an **auth_attr**(4) entry. Entries can come from any of the sources specified in the **nsswitch.conf**(4) file.

The **getauthattr( )** function enumerates **auth_attr** entries. The **getauthnam( )** function searches for an **auth_attr** entry with a given authorization name *name*. Successive calls to these functions return either successive **auth_attr** entries or **NULL**.

Th internal representation of an **auth_attr** entry is an **authattr_t** structure defined in **<auth_attr.h>** with the following members:

```
char    *name;/* name of the authorization */
char    *res1;/* reserved for future use */
char    *res2;/* reserved for future use */
char    *short_desc;/* short description */
char    *long_desc;/* long description */
kva_t   *attr;/* array of key-value pair attributes */
```

The **setauthattr( )** function "rewinds" to the beginning of the enumeration of **auth_attr** entries. Calls to **getauthnam( )** can leave the enumeration in an indeterminate state. Therefore, **setauthattr( )** should be called before the first call to **getauthattr( )**.

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

The **chkauthattr( )** function verifies whether or not a user has a given authorization. It first reads the **AUTHS_GRANTED** key in the **/etc/security/policy.conf** file and returns 1 if it finds a match for the given authorization. If **chkauthattr( )** does not find a match, it reads the **user_attr**(4) database. If it does not find a match in **user_attr**, **chkauthattr( )** reads the **prof_attr**(4) database, using the list of profiles assigned to the user, and checks if any of the profiles assigned to the user has the given authorization. **chkauthattr( )** returns 0 if it does not find a match in any of the three sources.

A user is considered to have been assigned an authorization if any of the following are true: 1) the authorization name matches exactly any authorization assigned in the **user_attr** or **prof_attr** databases (authorization names are case-sensitive), or 2) if the

authorization name suffix is not the key word **grant**, and the authorization name matches any authorization up to the asterisk (*) character assigned in the **user_attr** or **prof_attr** databases. The examples in the following table should make this more clear.

Authorization name	/etc/security/policy.conf or **user_attr** or **prof_attr** entry	Is user authorized?
com.sun.printer.postscript	com.sun.printer.postscript	YES
com.sun.printer.postscript	com.sun.printer.*	YES
com.sun.printer.grant	com.sun.printer.*	NO

The **free_authattr** function call releases memory allocated by the **getauthnam** and **getauthattr** functions.

RETURN VALUES

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

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

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

The **chkauthattr**(3) function returns 1 if the user is authorized and 0 otherwise.

CAVEATS

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

FILES

/etc/nsswitch.conf	stores configuration file lookup information for the name server switch
/etc/user_attr	stores extended user attributes
/etc/security/auth_attr	stores authorization attributes
/etc/security/policy.conf	stores policy definitions.
/etc/security/prof_attr	stores profile information

ATTRIBUTES

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

ATTRIBUTE TYPE	ATTRIBUTE VALUE
MT-Level	MT-Safe.

SEE ALSO

getexecattr(3), **getprofattr**(3), **getuserattr**(3), **auth_attr**(4), **nsswitch.conf**(4), **prof_attr**(4), and **user_attr**(4)

NOTES

getauthattr( ) and **getauthnam( )** both allocate memory for the pointers they return. This memory should be de-allocated with the **free_authattr( )** 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.