NAME

getprofent, setprofent, endprofent, getprofentbyname, free_profent – get Trusted Solaris 2.x user profile description

SYNOPSIS

cc [*flag* ...] *file* ... **–ltsoldb –ltsol –lnsl –lcmd** [*library* ...]

#include <**tsol/prof.h**>

profent_t *getprofentbyname(char *name, int src);

profent_t *getprofent(int src);

void setprofent(int stayopen, int src);

void endprofent(int src);

void free_profent(profent_t *profent);

AVAILABILITY

SUNWtsolu

MT-LEVEL

MT-Safe

DESCRIPTION

These functions are used to obtain entries describing Trusted Solaris user profiles from the **tsolprof** NIS+ database or the **/etc/security/tsol/tsoluser** file.

getprofentbyname() searches for information for a profile with the specified profile name given by the parameter *name*.

The functions **setprofent()**, **getprofent()**, and **endprofent()** are used to enumerate profile entries from the database. **setprofent()** sets (or resets) the enumeration to the beginning of the set of Trusted Solaris profile entries. This function should be called before the first call to **getprofent()**. A call to **getprofentbyname()** leaves the enumeration position in an indeterminate state. If the *stayopen* flag is non-zero, the system may keep allocated resources such as open file descriptors until a subsequent call to **endprofent()**.

Successive calls to **getprofent()** return either successive entries or return **NULL**, indicating the end of the enumeration.

endprofent() may be called to indicate that the caller expects to do no further profile entry retrieval operations; the system may then deallocate resources it was using. It is still allowed, but possibly less efficient, for the process to call more profile entry retrieval functions after calling **endprofent()**.

The functions **getprofentbyname()** and **getprofent()** are reentrant interfaces that allocate memory to store returned results, and are safe for use in both single-threaded and multithreaded applications. The function **free_profent()** should be used to free the pointers returned by either **getprofentbyname()** or **getprofent()**.

The parameter *name* must be a pointer to the profile name in the form of a null-terminated character-string.

The parameter *src* may be set to any of **TSOL_DB_SRC_FILES ,** **TSOL_DB_SRC_NISPLUS ,** or **TSOL_DB_SRC_SWITCH ,** which are defined in <**tsol/tsol.h**>. For most applications the *src* parameter should be set to **TSOL_DB_SRC_SWITCH ,** indicating that the system should use the **/etc/nsswitch.conf** file to determine the ultimate source of the database. However, in certain administrative application, it may be prudent to use the **TSOL_DB_SRC_FILES** option to for a read from the **/etc/security/tsol/tsoluser** file or the **TSOL_DB_SRC_NISPLUS** option to force a read from the tsoluser NIS+ database.

For enumeration in multithreaded applications, the position within the enumeration is a process-wide property shared by all threads. **setprofent()** may be used in a multithreaded application but resets the enumeration position for all threads. If multiple threads interleave calls to **getprofent()** , the threads will enumerate disjoint subsets of the **tsolprof database.**

RETURN VALUES

User entries are represented by the **struct profent_t** structure defined in <**tsol/prof.h**> **:**

```
typedef struct profent_s {
        char *          name;       /* name of profile */
        char *          desc;       /* description */
        char *          auths;      /* comma separated list of authorization numbers */
        profact_t *     actions;    /* linked list of actions */
        profcmd_t *     cmds;       /* linked list of commands */
} profent_t;
```

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

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

ERRORS

The functions **getprofentbyname()** and **getprofent()** return **NULL** on a failure.

NOTES

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.

When compiling multithreaded applications,see **Intro (3),** *Notes On Multithread Applications*, for information about the use of the **_REENTRANT** flag.