NAME

user_attr − Extended user attributes database

SYNOPSIS

/etc/user_attr

DESCRIPTION

/etc/user_attr is a local source of extended attributes associated with users and roles. **user_attr** can be used with other user attribute sources, including the **user_attr** NIS map and NIS+ table. Programs use the **getuserattr**(3) routines to gain access to this information.

The search order for multiple **user_attr** sources is specified in the **/etc/nsswitch.conf** file, as described in the **nsswitch.conf**(4) man page.

Each entry in the **user_attr** databases consists of a single line with five fields separated by colons (:). Line continuations using the backslash (\) character are permitted. Each entry has the form:

user:qualifier:res1:res2:attr

user The name of the user as specified in the **passwd**(4) database.

qualifier Reserved for future use.

res1 Reserved for future use.

res2 Reserved for future use.

attr An optional list of semicolon-separated (;) key-value pairs that describe the security attributes to apply to the object upon execution. Zero or more keys may be specified. There are four valid keys: **auths**, **profiles**, **roles**, **type**.

auths specifies a comma-separated list of authorization names chosen from those names defined in the **auth_attr**(4) database. Authorization names may be specified using the asterisk (*) character as a wildcard. For example, **solaris.printer.*** means all of Sun's printer authorizations.

profiles contains a ordered, comma-separated list of profile names chosen from **prof_attr**(4). Profiles are enforced by the profile shells, **pfcsh**, **pfksh**, and **pfsh**. See the **pfsh**(1)**man**page. If no profiles are assigned, the profile shells do not allow the user to execute any commands.

roles can be assigned a comma-separated list of role names from the set of user accounts in this database whose **type** field indicates the account is a role. If the **roles** key value is not specified, the user is not permitted to assume any role.

type can be assigned one of these strings: **normal**, indicating that this account is for a normal user, one who logs in; or **role**, indicating that this account is for a role. Roles can only be assumed by a normal user after the user has logged in.

EXAMPLE

The following example entry assigns to root the All profile, which allows root to use all commands in the system, and also assigns two authorizations:

root::::auths=solaris.*,solaris.grant;profiles=All;type=normal

The **solaris.*** wildcard authorization shown above gives root all the **solaris** authorizations; and the **solaris.grant** authorization gives root the right to grant to others any **solaris** authorizations that root has. The combination of authorizations enables root to grant to others all the **solaris** authorizations. See **auth_attr**(4) for more about authorizations.

With the following **nsswitch.conf** entry, the local **user_attr** file is consulted before the **user_attr** NIS+ table.

user_attr: files nisplus

FILES

/etc/nsswitch.conf

/etc/user_attr

CAVEATS

When deciding which authorization source to use (see **DESCRIPTION),** keep in mind that NIS+ provides stronger authentication than NIS.

The root user is usually defined in local databases for a number of reasons, including the fact that root needs to be able to log in and do system maintenance in single-user mode, before the network name service databases are available. For this reason, an entry should exist for root in the local **user_attr** file, and the precedence shown in the example **nsswitch.conf(4)** file entry under **EXAMPLE** is highly recommended.

Because the list of legal keys is likely to expand, any code that parses this database must be written to ignore unknown key-value pairs without error. When any new keywords are created, the names should be prefixed with a unique string, such as the company's stock symbol, to avoid potential naming conflicts.

In the **attr** field, escape the following symbols with a backslash (\) if you use them in any value: colon (:), semicolon (;), carrriage return (\n), equals (=), or backslash (\).

SEE ALSO

auths(1)**, auth_attr**(4)**, exec_attr**(4)**. getuserattr**(3)**, pfcsh**(1)**, pfksh**(1)**, pfsh**(1)**, prof_attr**(4)**, profiles**(1)**, roles**(1)